NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

UNITED STATES NAVY
NAVAL POSTGRADUATE SCHOOL

THESIS

ANALYSIS OF REAL TIME EMITTER LOCATION ALGORITHMS FOR TACTICAL ELECTRONIC WARFARE AIRCRAFT

by

Steven P. Jones

March 1998

Thesis Advisor:	Robert G. Hutchins
Second Reader:	Harold A. Titus

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE ANALYSIS OF REAL TIME EMITTER LOCATION ALGORITHMS FOR TACTICAL ELECTRONIC WARFARE AIRCRAFT

5. FUNDING NUMBERS

6. AUTHOR(S) Steven P. Jones

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Geographic location of radar emitters is the process of estimating an emitter's location upon the surface of the earth from direction of arrival (DOA) data for the targeted emitter. The current Emitter Location (EMLOC) algorithm utilized by the Grumman EA-6B Prowler is based on a thesis presented by Mr. Richard Opperman in June 1982. With the advent of increased processing demands on the AN/AYK-14 Tactical Computer as part of recent software upgrades to the AN/ALQ-99 Tactical Jamming System, it was hoped that a Kalman Filter, or Extended Kalman Filter based algorithm would reduce the processing time and memory requirements for the EMLOC algorithm. This thesis compares the current algorithm, and the Kalman/Extended Kalman Filters in a tactical scenario to determine if a change in the current Onboard Flight Program (OFP) should be recommended.

14. SUBJECT TERMS Kalman Filter, Extended Kalman Filter, Location Algorithm

15. NUMBER OF PAGES 87

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500
(Rev. 2-89)

Standard Form 298

Prescribed by ANSI Std. 239-18 298-102

ANALYSIS OF REAL TIME EMITTER LOCATION ALGORITHMS FOR TACTICAL ELECTRONIC WARFARE AIRCRAFT

Steven P. Jones
Major, United States Marine Corps
B.S., United States Naval Academy, 1986

Submitted in partial fulfillment
of the requirements for the degree of

**MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL
March 1998**

Author: ______________________________
Steven P. Jones

Approved by: Robert G Hutch
Robert G. Hutchins, Thesis Advisor

H A Titus
Harold A. Titus, Second Reader

____
Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

ABSTRACT

Geographic location of radar emitters is the process of estimating an emitter's location upon the surface of the earth from direction of arrival (DOA) data for the targeted emitter. The current Emitter Location (EMLOC) algorithm utilized by the Grumman EA-6B Prowler is based on a thesis presented by Mr. Richard Opperman in June 1982. With the advent of increased processing demands on the AN/AYK-14 Tactical Computer as part of recent software upgrades to the AN/ALQ-99 Tactical Jamming System, it was hoped that a Kalman Filter, or Extended Kalman Filter based algorithm would reduce the processing time and memory requirements for the EMLOC algorithm. This thesis compares the current algorithm, and the Kalman/Extended Kalman Filters in a tactical scenario to determine if a change in the current Onboard Flight Program (OFP) should be recommended.

TABLE OF CONTENTS

I. BACKGROUND 1

A. PURPOSE 1

B. AIRCRAFT 1

C. AN/ALQ-99 1

D. EMLOC FUNCTION 3

II. THE PROBLEM 5

A. APPLICATION 5

III. THE BOOZOO ALGORITHM 9

A. The BOOZOO ALGORITHM DEVELOPED 9

IV. THE ALTERNATIVES 17

A. THE KALMAN FILTER 17

B. THE EXTENDED KALMAN FILTER 21

V. THE SIMULATION 25

A. TESTED PARAMETERS 25

B. SCENARIO 25

VI. BOOZOO RESULTS 27

A. BOOZOO ALGORITHM 27

B. BASIC SCENARIO 28

C. EFFECTS OF SAMPLING INTERVAL 30

D. EFFECTS OF START/STOP POSITION 33

VII. KALMAN FILTER RESULTS 39

A. KALMAN FILTER 39

B. BASIC SCENARIO 40

C. EFFECTS OF SAMPLING INTERVAL 42

D. EFFECTS OF START/STOP POSITION 45

VIII. EXTENDED KALMAN FILTER RESULTS 51

A. THE EXTENDED KALMAN FILTER 51

B. BASIC SCENARIO 51

IX. CONCLUSIONS 53

A. EFFICIENCY 53

B. MEMORY REQUIREMENTS 53

C. ACCURACY 54

D. FUTURE DEVELOPMENTS 54

E. AREAS FOR FUTURE RESEARCH 55

APPENDIX A. BOOZOO CODE 57

APPENDIX B. COBURN KALMAN FILTER CODE 63

APPENDIX C. MILLS' KALMAN FILTER CODE 69

LIST OF REFERENCES 75

INITIAL DISTRIBUTION LIST 77

I. BACKGROUND

A. PURPOSE

This thesis examines current and proposed methods of geographic emitter location (EMLOC) with respect to the EA-6B aircraft via the AN/ALQ-99. The current algorithm will be compared to a proposed Kalman Filter based design. The various alternatives to EMLOC, along with their underlying assumptions, will be verified mathematically and a proposed test developed.

B. AIRCRAFT

The EA-6B aircraft is currently flown with three variants of the Improved Capabilities II (ICAP II) software/hardware upgrades. These are Blocks 82, 86, and 89A. The difference in the various blocks will be discussed, however, the focus of this paper will be Block 86/89A configured aircraft.

C. AN/ALQ-99

The AN/ALQ-99 is composed of two major sections. The Tactical Jamming System (TJS) carries out the majority of the Electronic Attack (EA) capability of the system. The Onboard System (OBS) is effectively the Electronic Support (ES) portion of the system. By utilizing the superhetrodyned receivers, the operator may scan various prioritized portions of the electromagnetic spectrum, or the entire effective frequency range for the receivers. For Block 82 aircraft, as receivers are scanned, they are controlled by the Command Mission Computer (CMC) and the Computer Interface Unit (CIU). If RF energy is received by any one of the receivers, that information is sent to the Encoder where information is digitized and sent to the CMC for further processing. Only one receiver's data may be processed at a time. The system, therefore, has high sensitivity, but low probability of intercept with eight bands scanned by six receivers, and only one channel for processing. Block 86/89A aircraft combine the encoder and CIU to form a CIU/E which also assists in matching emitters with known parameters. Additionally, two channels are provided to the CMC to speed (in theory, double) processing. Test data does not support increased processing speed, however, all aircraft will ultimately be deployed in the Block 89A configuration and thus this version will be the focus of this thesis.

As stated previously, the receivers are digitally tuned and superhetrodyned. Various methods are employed to prevent image frequencies from entering the system for further processing. However, these methods are not fool proof and operators must use care to employ hardware fixes, especially near high impulse density energy sources. Each receiver has 4-6 antennas arranged in the tail of the aircraft to localize the direction of arrival. For this paper we will focus upon the six antenna arrangement.

The six antennas are arranged at 60 degree increments to cover 360 degrees of space around the aircraft. As signals are collected by the receivers, the amplitudes are compared at the different feeds, and a geometric average is sent to the encoder. This method of amplitude comparison is not very accurate, but in theory would work as follows:

> Assume at feed 1 the amplitude was 2 units.
>
> At feed 2 (clockwise from feed 1) the amplitude was 1 unit.
>
> Therefore, the energy is twice as great at feed 1, so the Direction of Arrival (DOA) is closer to feed 1, but how much closer?
>
> By taking a weighted average, the signal DOA must be 20 degrees to the right of feed 1 and 40 degrees to the left of feed 2.
>
> Thus if feed one is boresighted with the aircraft heading, the DOA is 20 degrees relative to the aircraft nose in the direction of feed 2 boresight.

In theory, this system provides quick, accurate results, with inexpensive hardware. Experience, however, shows that the six antenna system achieves a nominal +/-10 degree accuracy. Additionally, a DOA does not guarantee sure location of the detected emitter. Atmospherics (refraction) and multi-path (reflections from ground obstacles) contribute bogus DOAs. Finally, if energy is detected at three feeds (which is possible with close proximity high powered signals), the detection is dropped and the receiver continues to its next programmed tuning step.

D. EMLOC FUNCTION

When a receiver detects a threat (or friendly for that matter) emitter and a match is made with known parameters, it is displayed to the operator in frequency and azimuth (after degrees true conversion). The operator may "hook" the displayed symbol and via several software steps, select the target emitter for Emitter Location, or EMLOC. The

OBS then runs each subsequent detection of this emitter match through an algorithm that attempts to calculate the geographic location of the emitter.

The actual mathematics necessary to accomplish this task are not terribly difficult to compute, but an explanation is necessary. The following chapter will describe the geometry and basic assumptions required to compute a target emitter's geographic location given a change in aircraft position and a change in DOA.

II. THE PROBLEM

A. APPLICATION

All DOAs are delivered to the OBS relative to the aircraft. The results need only be developed relative to the aircraft since navigation is now precise with the advent of Global Positioning System (GPS) in Block 89A. Thus, with an emitter position calculated relative to the aircraft position (which is known with some precision) the emitter may be geo-located. The triangulation of emitters via Direction Finding (DF) equipment may be accomplished manually, and the "Flat Earth Approximation" geometry is developed quite simply (see Figure 1).

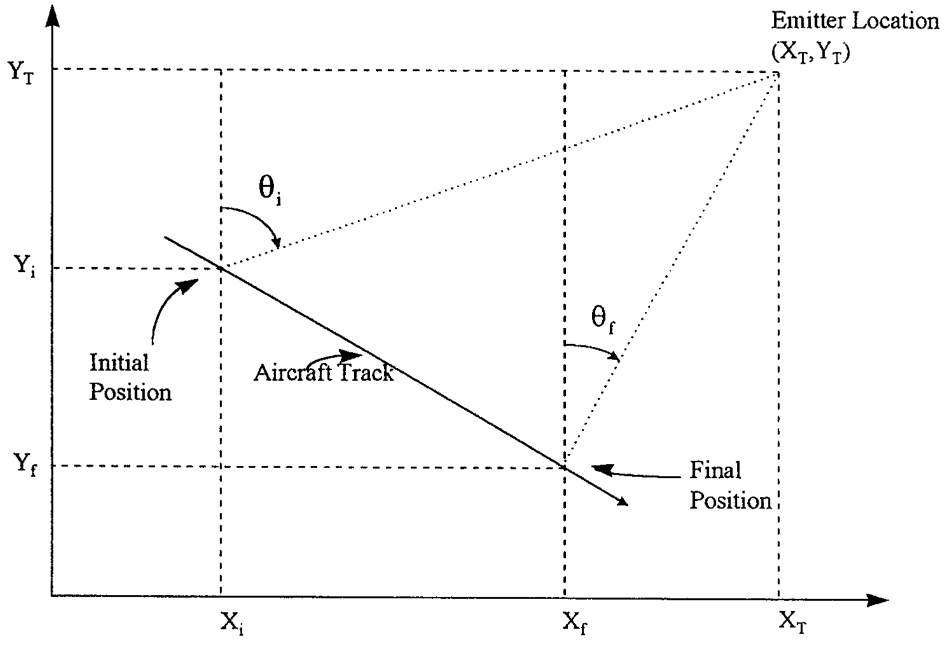

Figure 1: Flat Earth Approximation Geometry

$$\tan\theta_i = \frac{X_T - X_i}{Y_T - Y_i} \tag{2-1}$$

and

$$\tan\theta_f = \frac{X_T - X_f}{Y_T - Y_f} \tag{2-2}$$

If latitude and longitude are substituted for X-Y axes, the following equations result:

$$\tan\hat{\theta}_i = \frac{(\lambda_T - \lambda_i)\cos L_T}{L_T - L_i} \tag{2-3}$$

and

$$\tan\hat{\theta}_f = \frac{(\lambda_T - \lambda_f)\cos L_T}{L_T - L_f} \tag{2-4}$$

where the 'i' subscript indicates the aircraft initial position at detection and match, and the 'f' subscript indicates the final aircraft position. The emitters coordinates are (X_T, Y_T) are unknown. These coordinates can be further transcribed into geographic latitude and longitude, but we will not pursue these calculations at this time.

More importantly, by examining the flat earth approximation we will validate these "simple" equations, vice more complicated spherical solutions required by long range systems. Since we are only concerned with microwave frequencies, we can approximate the maximum distance for direct Electromagnetic propagation with:

$$d = 1.23\left(\sqrt{h_a} + \sqrt{h_e}\right)$$

where d = radar horizon in nautical miles,

$h_a, h_e =$ altitude in feet for the aircraft and emitter

assuming the 4/3 earth radius approximation for refraction characteristics. Thus if h_e is near sea level and aircraft ceiling is less than 45,000 feet, radar horizons should not exceed 261 nautical miles (300 statute miles). L.Laddie Coburn [Ref.1] proved that at distances less than 360 nautical miles the planar solution was less than 0.5% in error when compared to the exact spherical solution, and that for ranges less than 150 nautical miles, the planar solution was sufficiently accurate for emitter location estimation.

Unfortunately, the EMLOC problem is not quite this simple. With refractive errors and multipath, there are many arriving DOAs to consider. Also, the simple geometry shown in Figure 1 is not always so clear. Typical changes in bearing may only be 2-4 degrees over several minutes flying time when aircraft aspect is nose/tail and/or range to target emitter is great. Therefore, various algorithms are applied that employ maximum likelihood estimators to locate emitters and thus, most geographic locations are a statistical approximation of the most probable location of the emitter.

The algorithm currently employed by the EA-6B is an iterative method that was developed by Daniel Charles Opperman in June 1977. He developed his method and refers to it as the BOOZOO algorithm (not the term most operators would have preferred). During the early 1970's, several students at the Naval Postgraduate School, under the tutelage of Professor Harold Titus investigated the use of Kalman Filters as a possible method for geo-locating emitters. We will develop both methods and evaluate their performance as to accuracy, processing speed and robustness under less than optimum geometry.

III. THE BOOZOO ALGORITHM

A. THE BOOZOO ALGORITHM DEVELOPED

As stated previously, the Boozoo Algorithm (BZA) was developed by Opperman as part of his thesis work with the Tactical Electronic Reporting, Processing, and Evaluation System (TERPES) [Ref. 2]. This system was developed by the U.S. Marine Corps to take ES data collected by a variety of sources (primarily the EA-6B) and develop the Electronic Order of Battle (EOB). The system is currently fielded and is organic to the Marine Tactical Electronic Warfare Squadrons. Opperman developed his thesis based upon equations provided by the North American Rockwell Corporation (NAR). Opperman noted that the NAR equations were based on a model where the ES aircraft flew a track long enough to create at least 90 degrees of change in bearing. This is not always possible so the BZA was developed to account for DOA error that was significant when compared to the change in the initial bearing and final bearing. The BZA was eventually incorporated into the TERPES system in the mid 1980's and into the EA-6B aircraft during the early 1990's with the EMLOC function.

Opperman used the following as a foundation:

$$P(\alpha_i)d\alpha_i = \frac{\exp\left(-\alpha_i^2 \big/ 2\sigma_i^2\right)}{\sqrt{2\pi}\sigma_i} d\alpha_i \qquad (3\text{-}1)$$

where $P(\alpha_i)$ is the probability that the ith aircraft position will register a bearing between $\theta_i + \alpha_i$ and $\theta_i + \alpha_i + d\alpha_i$, and α_i is the DOA measurement error for the *i*th measurement. The value, σ_i is the standard deviation of the DOA bearing error associated with the ith measurement.

If we take n independent measurements, then we have joint probability:

$$P(\alpha_1,\ldots,\alpha_n)d\alpha_1 d\alpha_2\ldots d\alpha_n = \frac{\exp\left(-1/2f\left(\hat{E}\right)\right)}{\left((2\pi)^{n/2}\right)\sigma_1\sigma_2\ldots\sigma_n}d\alpha_1 d_2\ldots d\alpha_n \tag{3-2}$$

where

$$f\left(\hat{E}\right) = \sum_{i=1}^{n}\left(\alpha_i / \sigma_i\right)^2 \tag{3-3}$$

and (see figure 2) $\hat{E}$ is the most likely position of the target emitter, $\hat{r}$ is the aircraft position vector, and $\hat{t}$ is a vector orthogonal to $\hat{r}$, and lying in the plane defined by the vectors, Q, (actual position vector) and $\hat{E}$,

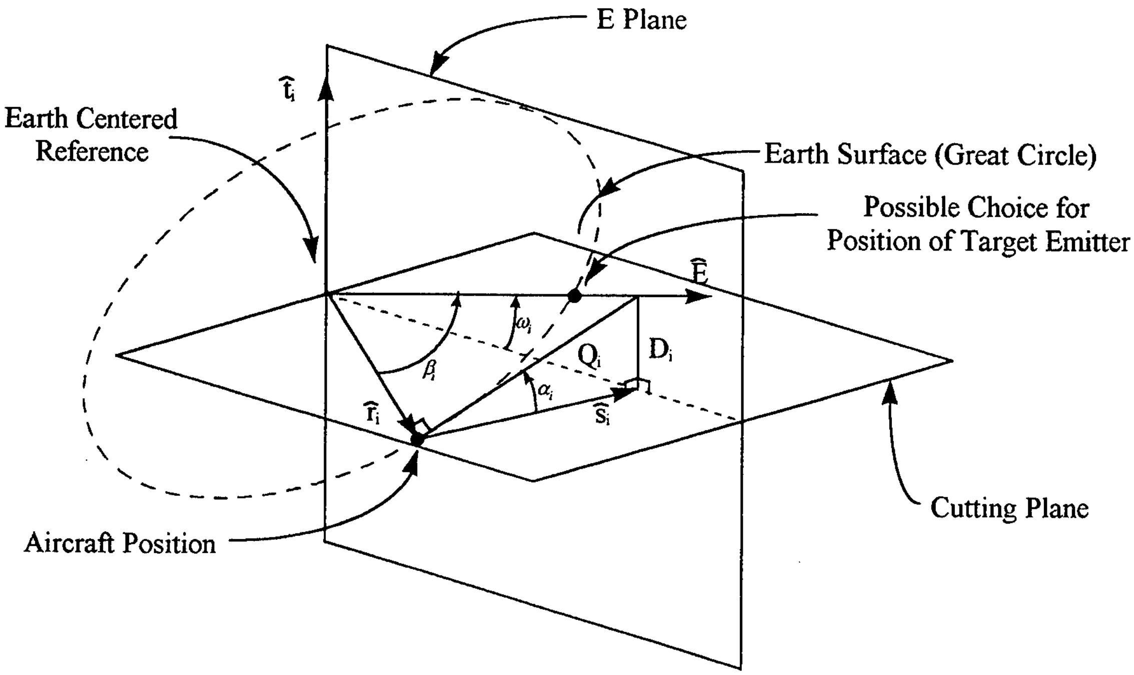

Figure 2: Unit Sphere Approximation of Emitter Location

In order to minimize the error, Opperman suggested a weighting function that would be changed according to where the DOA came from. In our case this would imply some DOAs are more accurate than others. This turns out to be true in actuality and therefore a weighting function is defined:

$$W_i = {}^{1}\!\big/_{k\sigma_i^2} \tag{3-4}$$

where *k* is an arbitrary constant of proportionality.

Thus the probability would be maximized if we minimize

$$f\left(\hat{E}\right) = k\sum_{i=1}^{n} W_i^2 \alpha_i^2 \tag{3-5}$$

As part of his foundation, Opperman does this, by first making an approximation. If α_i is small then $\sin\alpha_i \approx \alpha_i$ so,

$$f\left(\hat{E}\right) = k\sum_{i=1}^{n} W_i^2 \sin^2\alpha_i \tag{3-6}$$

From Figure 2,

$$\sin\alpha_i \approx \cos\left({}^{\pi}\!\big/_{2} - \omega_i\right) / \sin\beta_i$$

so now,

$$f\left(\hat{E}\right) = k\sum_{i=1}^{n} W_i^2\left(\cos^2\left({}^{\pi}\!\big/_{2} - \omega_i\right)\big/\sin^2\beta_i\right) \tag{3-7}$$

We may assume that

$$\sin^2\beta_i \approx \frac{1}{n}\sum_{i=1}^{n}\sin^2\beta_i$$

and making another substitution with simplification,

$$f\left(\hat{E}\right) = nk \frac{\sum_{i=1}^{n} W_i^2 \cos^2\left(\pi/_2 - \omega_i\right)}{\sum_{i=1}^{n} \sin^2 \beta_i} \tag{3-8}$$

Equation (3-8) when viewed with Figure 2 may be expressed in terms of the expected emitter position, $\hat{E}$.

$$f\left(\hat{E}\right) = nk \frac{\sum_{i=1}^{n} W_i^2 \left(\hat{E} \cdot \hat{t}_i\right)^2}{\sum_{i=1}^{n} \left|\hat{E} \times \hat{r}_i\right|^2} \tag{3-9}$$

The function in equation (3-9) may be minimized, but some other reference may need to be defined. Assuming λ and ϕ are longitude and latitude of the aircraft position respectively, and θ remains the measured DOA as measured from true north, we may redefine the vectors, $\hat{E}, \hat{r}$, and $\hat{t}$ such that:

$$\begin{aligned}
a_i &= \cos\lambda_i \cos\phi_i \\
b_i &= \sin\lambda_i \cos\phi_i \\
c_i &= \sin\phi_i \\
X_i &= \sin\lambda_i \cos\theta_i - \sin\phi_i \cos\lambda_i \sin\theta_i \\
Y_i &= -\cos\lambda_i \cos\theta_i - \sin\phi_i \sin\lambda_i \sin\theta_i \\
Z_i &= \cos\phi_i \sin\theta_i
\end{aligned}$$

Where. Finally, we use earth centered coordinates (i,j,k) to define:

$$\begin{aligned}
\hat{E} &= u\hat{i} + v\hat{j} + w\hat{k} \\
\hat{t}_i &= X_i\hat{i} + Y_i\hat{j} + Z_i\hat{k} \\
\hat{r}_i &= a_i\hat{i} + b_i\hat{k} + c_i\hat{k}
\end{aligned}$$

Now substituting into equation (3-9)

$$f\left(\hat{E}\right) = g / h \tag{3-10}$$

where

$$\begin{aligned} g &= nk\sum_{i=1}^{n} W_i^2\left(X_i u + Y_i v + Z_i w\right)^2 \\ h &= \sum_{i=1}^{n}\left[\left(b_i w - c_i v\right)^2 + \left(c_i u - a_i w\right)^2 + \left(a_i v - b_i u\right)^2\right] \end{aligned}$$

By differentiating with respect to u, v, and w we obtain the gradient of the function.

$\nabla f\left(f\left(\hat{E}\right)\right)$ will have three components

$$\begin{aligned} f_u &= \left(g_u - h_u f\left(\hat{E}\right)\right)/h \\ f_v &= \left(g_v - h_v f\left(\hat{E}\right)\right)/h \\ f_w &= \left(g_w - h_w f\left(\hat{E}\right)\right)/h \end{aligned} \tag{3-11}$$

Now, if the gradient is allowed to approach zero, then $f\left(\hat{E}\right)$ will be minimized and equations (3-11) will become

$$\begin{aligned} g_u &= h_u f\left(\hat{E}\right) \\ g_v &= h_v f\left(\hat{E}\right) \\ g_w &= h_w f\left(\hat{E}\right) \end{aligned} \tag{3-12}$$

By differentiating functions g and h and substituting into equations (3-12) we obtain:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = f\left(\hat{E}\right)\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3-13}$$

where

$$a_{11} = \sum_{i=1}^{n} X_i^2 W_i^2 \qquad a_{22} = \sum_{i=1}^{n} Y_i^2 W_i^2$$
$$a_{12} = \sum_{i=1}^{n} X_i Y_i W_i^2 \qquad a_{23} = \sum_{i=1}^{n} Y_i Z_i W_i^2$$
$$a_{13} = \sum_{i=1}^{n} X_i Z_i W_i^2 \qquad a_{33} = \sum_{i=1}^{n} Z_i^2 W_i^2$$

and

$$b_{11} = \sum_{i=1}^{n} \left(b_i^2 + c_i^2\right) \qquad b_{22} = \sum_{i=1}^{n} \left(a_i^2 + c_i^2\right)$$
$$b_{12} = -\sum_{i=1}^{n} a_i b_i \qquad b_{23} = -\sum_{i=1}^{n} b_i c_i$$
$$b_{13} = -\sum_{i=1}^{n} a_i c_i \qquad b_{33} = \sum_{i=1}^{n} \left(a_i^2 + b_i^2\right)$$

The solution for *u*, *v*, and *w* is simplified if we assume that the right side of equation (3-13) has a known value. Therefore by using a method of iteration we obtain:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} u_{k+1} \\ v_{k+1} \\ w_{k+1} \end{bmatrix} = f\left(\hat{E}\right) \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{bmatrix} \begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix} \tag{3-14}$$

where $k = 0,1,2,3,...m$, and the zeroth iteration of *u*, *v*, and *w* are the solutions for the emitters position.

Simplifying the right half of equation (3-14),

$$\underline{A} \begin{bmatrix} u_{k+1} \\ v_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} u_{k+1} \\ v_{k+1} \\ w_{k+1} \end{bmatrix} = f\left(\hat{E}\right) \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \tag{3-15}$$

where

$$B_1 = b_{11} u_k + b_{12} v_k + b_{13} w_k$$
$$B_2 = b_{12} u_k + b_{22} v_k + b_{13} w_k$$
$$B_1 = b_{13} u_k + b_{23} v_k + b_{33} w_k$$

Equation (3-15) can be solved by Cramer's Rule, provided the A matrix is singular. Then,

$$\begin{aligned}
u_{k+1} &= \frac{f\left(\hat{E}\right)}{\det A}\left(\det\begin{vmatrix} B_1 & a_{12} & a_{13} \\ B_2 & a_{22} & a_{23} \\ B_3 & a_{23} & a_{33} \end{vmatrix}\right) \\
v_{k+1} &= \frac{f\left(\hat{E}\right)}{\det A}\left(\det\begin{vmatrix} a_{11} & B_1 & a_{13} \\ a_{12} & B_2 & a_{23} \\ a_{13} & B_3 & a_{33} \end{vmatrix}\right) \\
w_{k+1} &= \frac{f\left(\hat{E}\right)}{\det A}\left(\det\begin{vmatrix} a_{11} & a_{12} & B_1 \\ a_{12} & a_{22} & B_2 \\ a_{13} & a_{23} & B_3 \end{vmatrix}\right)
\end{aligned} \tag{3-16}$$

This solution, however, leads to the determination of the position vector in both direction and magnitude. Since we only need direction, we may further simplify the solution.

$$\begin{aligned}
u_{k+1} &= \left(\det\begin{vmatrix} B_1 & a_{12} & a_{13} \\ B_2 & a_{22} & a_{23} \\ B_3 & a_{23} & a_{33} \end{vmatrix}\right) \\
v_{k+1} &= \left(\det\begin{vmatrix} a_{11} & B_1 & a_{13} \\ a_{12} & B_2 & a_{23} \\ a_{13} & B_3 & a_{33} \end{vmatrix}\right) \\
w_{k+1} &= \left(\det\begin{vmatrix} a_{11} & a_{12} & B_1 \\ a_{12} & a_{22} & B_2 \\ a_{13} & a_{23} & B_3 \end{vmatrix}\right)
\end{aligned} \tag{3-17}$$

The remaining calculations are quite simple to find the latitude and longitude of the emitter position vector, $\hat{E}$.

$$\mathit{Longitude} = \arctan\left(v_m / u_m\right)$$

$$\mathit{Latitude} = \arctan\left(w_m \Big/ \sqrt{u_m^2 + v_m^2} \right)$$

The BZA appears to be a math intensive algorithm that would require an inordinate amount of processing time when compared to other methods of estimation. It is now appropriate to develop several alternative estimators to the same problem, the Kalman Filter, and Extended Kalman Filter.

IV. THE ALTERNATIVES

A. THE KALMAN FILTER

The Kalman Filter equations are a mathematical representation of linear systems and are adaptive to a variety of situations when trying to simulate, or estimate linear systems. They have extension to nonlinear systems, but may not be as accurate in certain circumstances. The application of Kalman Filter equations to the EMLOC problem have been explored since the early 1970's. In June, 1972, L. Laddie Coburn [Ref. 1] developed two algorithms, a Kalman Filter and an Extended Kalman Filter, since there are nonlinearities. Also of use was the follow-on work to Coburn by Edward H. Mills [Ref. 3] that was useful in creating algorithms compatible with the AYK-14. We will examine both algorithms.

The underlying mathematical model for the Kalman Filter tracker consists of a state equation (a linear difference equation) and a linear measurement equation. Noise processes appear in both equations.

The state equation is:

$$X(k+1) = \Phi(k+1|k)X(k) + \Gamma W(k) \tag{4-1}$$

where $\Phi(k+1|k)$ is the state transition matrix, Γ is a distribution matrix related to the random forcing function, $W(k)$ is a random forcing function that accounts for random excitations (noise), and k is the time step.

The measurement equation is:

$$Z(k) = H(k)X(k) + V(k) \qquad k = 1, 2, 3, \ldots \tag{4-2}$$

where $H(k)$ is the observation matrix, and $V(k)$ is the measurement noise.

The noise is assumed to have zero mean and therefore,

$$\begin{aligned} &E\left[V(k)V(j)^T\right] = R(k)\delta(kj) \\ &\Gamma E\left[W(k)W(j)^T\right]\Gamma^T = Q(k)\delta(kj) \\ &E\left[V(k)W(j)^T\right] = 0, \forall k, j \end{aligned} \tag{4-3}$$

where $\delta(kj)$ is the Kronecker delta function and has value zero unless $k = j$.

The Kalman recursion equations are summarized:

$$\begin{aligned} &G(k) = P(k|k-1)H(k)^T\left[H(k)P(k|k-1)H(k)^T + R(k)\right]^{-1} \\ &P(k|k) = P(k|k-1) - G(k)H(k)P(k|k-1) \\ &P(k+1|k) = \Phi(k+1,k)P(k|k)\Phi(k+1,k)^T + Q(k) \\ &\hat{X}(k|k) = \hat{X}(k|k-1) + G(k)\left[Z(k) - H(k)\hat{X}(k|k-1)\right] \\ &\hat{X}(k|k-1) = \Phi(k,k-1)\hat{X}(k-1|k-1) \end{aligned} \tag{4-4}$$

where $\hat{X}(k|j)$ is the estimate of state $X(k)$ based on j measurements ($Z(1)$, $Z(2)$,...$Z(j)$). The Kalman filter gains are represented by matrix $G(k)$ and $P(k|j)$ represents the error covariance matrix of the estimates.

Since we are concerned with DOAs and bearing rates, we will filter noisy observed DOAs to obtain an optimal estimate. This is best approximated by a state transition matrix:

$$\begin{aligned} &\Phi(k+1,k) = \begin{bmatrix} 1 & T(k+1) \\ 0 & 1 \end{bmatrix} \\ &H(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned} \tag{4-5}$$

Also, since DOAs are the only observed value, $R(k)$ and $W(k)$ become scalars and the bracketed terms in equations (4-4) become scalars, also. This will play a part in simplifying and reducing the matrix equations to scalar form. The non-uniformity of arrival time for DOAs requires that the term $T(k+1)$ remain a variable in each of the applicable recursion equations. This requires that the error covariance and gains be computed continuously since they are a function of the sample interval.

The P and Q matrices have scalar components given by

$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$ and the G matrix is a 2 x 1 matrix, $\begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$.

Rewriting the gain equation $G(k)$,

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left\{ \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R \right\}^{-1} \tag{4-6}$$

The observation matrix $H(k) = [1\ 0]$ forces the inverse term to become a scalar, allowing us to compute gain terms directly:

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} [P_{11} + R]^{-1} \tag{4-7}$$

which results in scalar gain equations

$$\begin{aligned} G_1(k) &= \frac{P_{11}(k|k-1)}{P_{11}(k|k-1) + R(k)} \\ G_2(k) &= \frac{P_{21}(k|k-1)}{P_{11}(k|k-1) + R(k)} \end{aligned} \tag{4-8}$$

Now the first DOA is not filtered, but a technique of filtering backward is available to improve the estimate of the initial DOA. These equations are similar to the Kalman equations, but they update and smooth previous data based on the last DOA that has been optimally estimated. This method was proposed by Rauch [Ref. 4] in 1963.

$$\begin{aligned} \hat{X}(1|k) &= \hat{X}(1|k-1) + D(1|k)G(k)\left[Z(k) - H(k)\hat{X}(k|k-1)\right] \\ D(1|k) &= D(1|k-1)P(k-1|k-1)\Phi(k,k-1)^T P(k|k-1)^{-1} \end{aligned} \tag{4-9}$$

where $D(1|k)$ becomes a function of $T(k)$ and must be solved sequentially before the

$$\bar{\theta}(k) = H(k)\bar{\Theta}(k) + V(k) \tag{4-10}$$

where $\mathrm{D}(1|k)$ becomes a function of $\mathrm{T}(k)$ and must be solved sequentially before the smoothed estimate can be computed. These equations can also be reduced to scalar form and were utilized by Coburn in his thesis work.

Initialization of the filter requires knowledge of the system dynamics, statistical properties of the filter, and information regarding the initial state of the system.

The noisy observation of the state may be made by

$$\bar{\theta}(k) = \begin{bmatrix} \theta(k) \\ \dot{\theta}(k) \end{bmatrix}$$

$$\bar{\Theta}(k) = \begin{bmatrix} \Theta(k) \\ \dot{\Theta}(k) \end{bmatrix}$$

where $\bar{\theta}(k)$ is a vector of noisy observations, and $\bar{\Theta}(k)$ is a state vector of exact emitter bearing angle and bearing rate. The observation matrix $H(k) = [1\ 0]$ is for measurement of the bearing angle only. Bearing rate depends on speed, range and heading of the aircraft with respect to the emitter. Since the range is initially unknown, it must be estimated. This could be a standardized variable based on the emitter to be located, or could be an operator input. For this exercise it will be denoted by the variable *r*. Therefore,

$$\dot{\theta} = \frac{v}{r}\sin(\text{Relative DOA})\left(\frac{57.29578^{0}\!\big/\!_{rad}}{3600}\right)\text{deg/ sec} \qquad (4\text{-}11)$$

The remaining equations are simple substitutions of theta for *X* in the basic Kalman equations presented, resulting in the optimal estimation equation:

$$\begin{bmatrix}\hat{\theta}(k|k)\\ \hat{\dot{\theta}}(k|k)\end{bmatrix} = \begin{bmatrix}1 & T(k)\\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta(k-1|k-1)\\ \hat{\dot{\theta}}(k-1|k-1)\end{bmatrix} + \begin{bmatrix}G_1(k)\\ G_2(k)\end{bmatrix}E(k) \qquad (4\text{-}12)$$

where

$$E(k) = \theta(k) - \hat{\theta}(k-1|k-1) - T(k)\hat{\dot{\theta}}(k-1|k-1)$$

Again, since there is not a uniform sampling interval, the filter gains *G* may vary and not approach a steady state value uniformly.

Once the values for theta are determined, an iterative approach to the equations in chapter two should lead to the emitter position.

B. THE EXTENDED KALMAN FILTER

Due to the nonlinearities of the trigonometric functions, theoretically an Extended Kalman Filter (EKF) is warranted for our situation. Realistically, we will not have *a* *priorii* bearing rate, nor will our initial guesses at bearing rate be accurate. Therefore, we must face reality and attempt to locate with only DOA data. If this is the case, however, a nonlinear transformation must be utilized to transform observable DOAs into filtered position estimates.

This is not as intimidating as it may sound, using the Extended Kalman Filter (EKF) with the help of small perturbation theory and a Taylor Series about an initial point. Equation (4-1) then becomes

$$Z(k) = N[X(k)] + V(k) \tag{4-13}$$

where $Z(k)$ represents observable DOAs, $X(k)$ is the emitter position vector, and N represents the nonlinear transformation

$$\hat{\theta} = \arctan\left(\frac{(\lambda_T - \lambda)\cos L_T}{L_T - L}\right) \tag{4-14}$$

If the position error of the state vector $X(k)$ is given by

$$\tilde{X}(k) = X(k) - \hat{X}(k), \tag{4-15}$$

then the true position $X(k)$ may be expanded about the most recent optimal estimate, $\hat{X}(k)$ in a Taylor series expansion with higher order terms thrown away as shown:

$$N[X(k)] = N[\hat{X}(k)] + M(k)\tilde{X} + \ldots \tag{4-16}$$

where

$$M(k) = \left.\frac{\partial N}{\partial X}\right|_{\hat{X}} = \left[{}^{\partial\hat{\theta}}\!/_{\partial\lambda_T} \quad {}^{\partial\hat{\theta}}\!/_{\partial L_T} \right]$$

for the emitter location algorithm given in equations (2-3) and (2-4).

The Kalman filter recursion equations become

$$\begin{aligned}
&G(k) = P(k|k-1)M(k)^T\left[M(k)P(k|k-1)M(k)^T + R(k)\right]^{-1}\\
&P(k|k) = P(k|k-1) - G(k)M(k)P(k|k-1)\\
&P(k+1)|k) = \Phi(k+1,k)P(k|k)\Phi(k+1,k)^T + Q(k)
\end{aligned} \tag{4-17}$$

Since the emitter is almost always stationary (or near stationary), the Φ matrix becomes the identity matrix, greatly simplifying the equations.

EKF filters can be volatile, and without proper initialization will diverge. Obviously, if the equations can not be initialized a majority of the time with good results, the EKF will be of no use to the operator.

Now that the various algorithms, both current and proposed, have been described mathematically, it is time to subject each to the same scenarios and determine which, if any, is clearly more reliable, or efficient.

V. THE SIMULATION

A. TESTED PARAMETERS

The ultimate goal of this project is to program the three algorithms presented to determine which method:

1.) is more computationally efficient (fewest floating point operations, or flops)

2.) requires least amount of memory (smallest matrices and arrays held in RAM)

3.) converges to an acceptable range of the target emitter in the fewest number of iterations.

B. SCENARIO

A realistic simulation was developed to test the algorithms using Monte Carlo techniques. All computer code was written in MATLAB.

The simulated ES aircraft flies North along the Prime Meridian (0 deg Longitude) from 1.5 deg S. Latitude to 1.5 deg N. Latitude, a distance of 180 Nautical Miles (Nmi). The aircraft will flies at constant altitude and heading at a true airspeed of 360 knots. Measurements are taken at 12 second intervals in an attempt to geo-locate the target emitter. The true emitter location is on the equator (0 deg latitude) at 1.5 deg East Longitude Direction of arrival accuracy was initially assumed to be +/- 10 degrees. A representation of each algorithm is included in the Appendices.

Also, in an effort to determine if geometry and/or sampling interval has an effect on any algorithm's performance, the start and stop points were altered.

The algorithms were tested as described above, and a chapter is devoted to the results of each.

VI. BOOZOO RESULTS

A. BOOZOO ALGORITHM

The conversion of Opperman's code was completed with the assistance of Mr. Lester Hathaway, NAWC Weapons Division, NAS Pt. Mugu, CA. Additionally, the actual weighting scheme for one of the frequency bands was employed. Figure 3 is a graphic representation of the weights applied to each arriving signal. Although every effort was made to economize the volume of computer code written in support of the algorithm, the AN/AYK-14 does not support matrix operations and must therefore rely

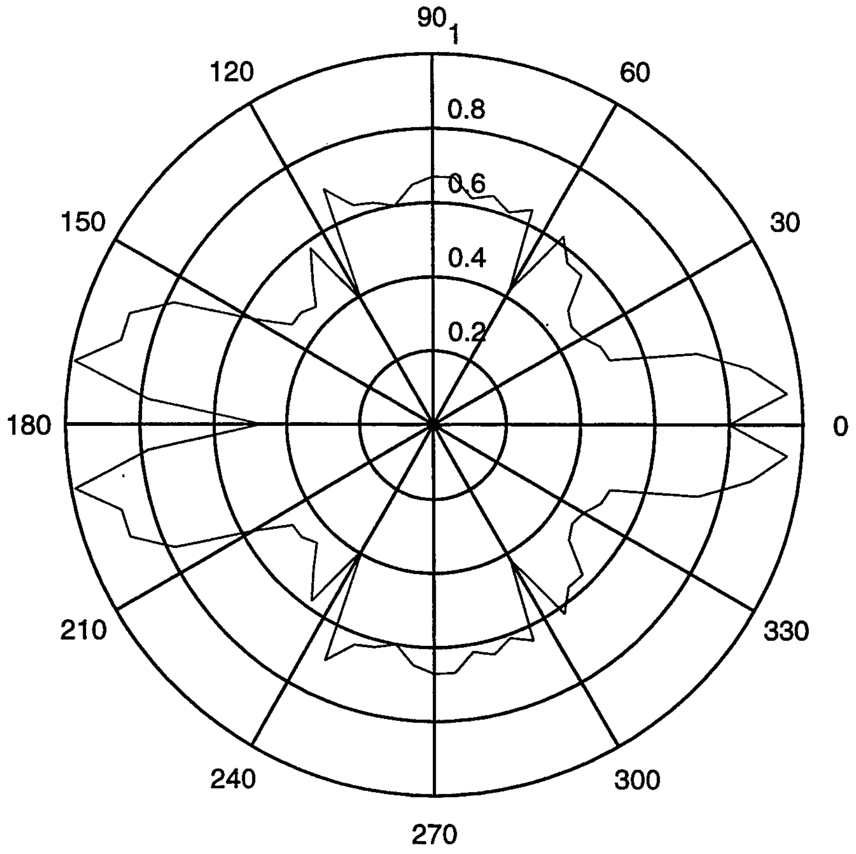

Figure 3: EA-6B Mid-Frequency Weighting Scheme

upon scalar equations. As performed for this simulation, the BZA requires memory positions for 75 variables, including 37 weights from the weight look-up table. An example of the code utilized is included as Appendix A.

B. BASIC SCENARIO

Figure 4, depicts a typical ES track along the Prime Meridian and the cluster of estimated positions for the target emitter for one run of the simulation. Accuracy was impressive as shown in Figure 5 for both one simulation run, and the average of 1000 simulations. The number of iterative loops per DOA cut was constant or nearly constant at two iterative loops with each loop comprising 42 floating point operations. Set up for the iterative process took an additional 68 floating point operations for a total of 152

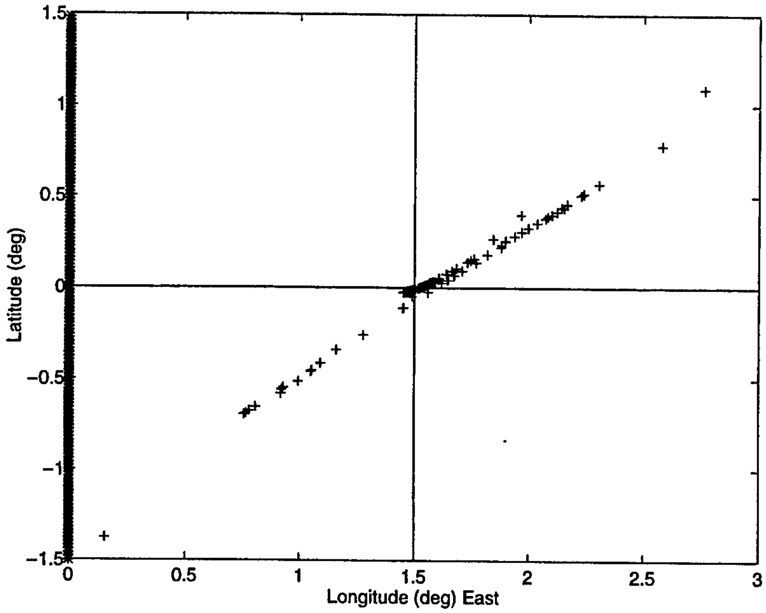

Figure 4: Single Run of BOOZOO Baseline Simulation

floating point operations per typical DOA cut. Finally, a simple time analysis was accomplished by evaluating the time to complete one thousand iterations of the complete scenario. The one thousand simulations took 2,256.2 seconds to complete, for an average of 2.5262 seconds for each loop through the scenario. Thus, with 150 measurements taken, the BZA process averages 0.0166 seconds per measurement to arrive at a position estimate. Figure 5 depicts the average error associated with the BZA under the conditions set in Chapter V. Figure 6 (shown on the next page) is a blown up depiction of average error focusing upon the area of convergence for the targeted emitter.

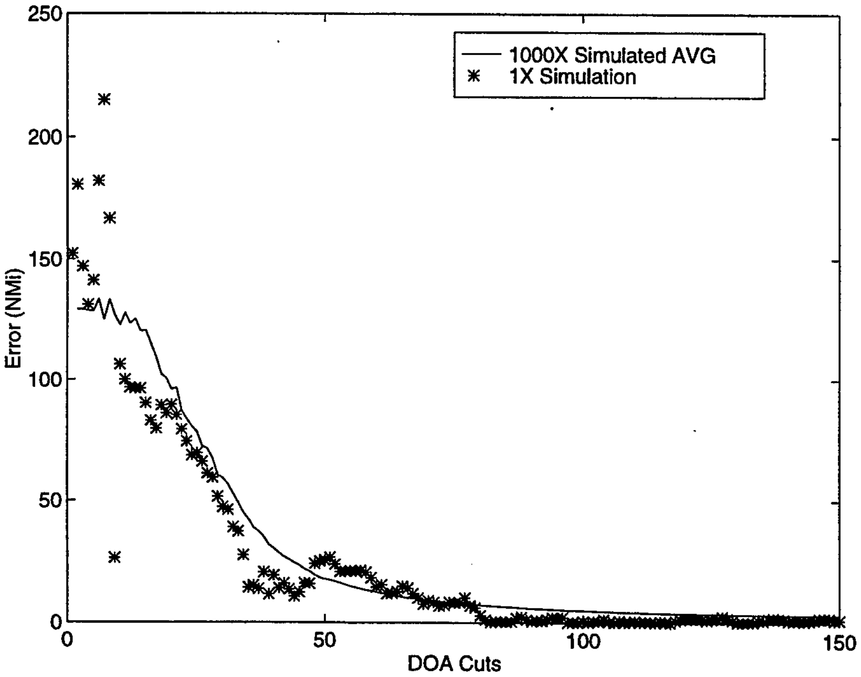

Figure 5: Averaged Error 1000 Simulations

The downward slope of the average error is very steep starting at about 50 measurements, and flattens out after 75 measurements, which corresponds with the aircraft passing the abeam position to the target emitter. Measurements were terminated at 150, which corresponds to a theoretical 90 degree change in target position. Final average range error is less than four nautical miles (Nmi.) at the termination of the test.

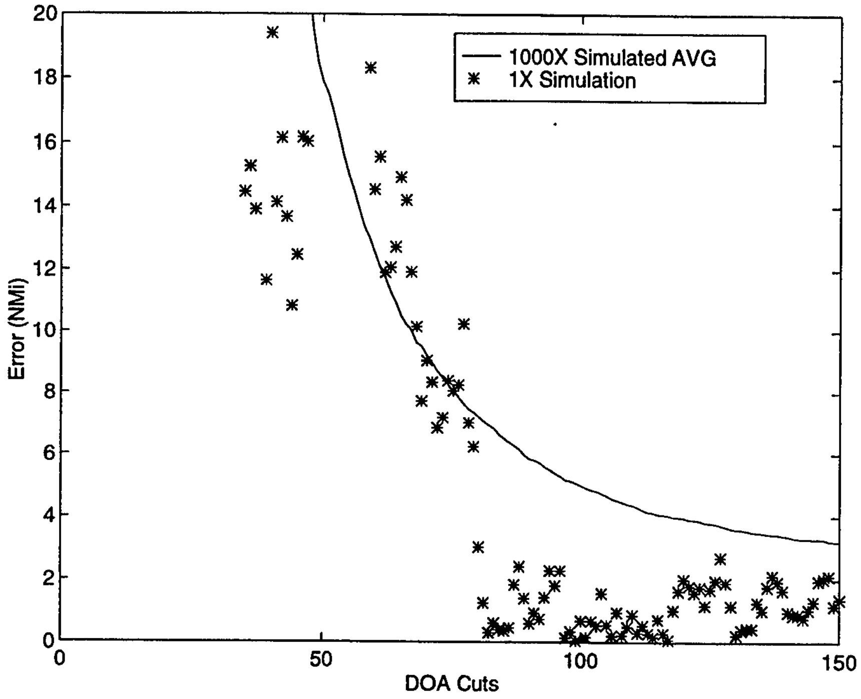

Figure 6: Averaged Error (<20 Nmi) 1000 Simulations

C. EFFECTS OF SAMPLING INTERVAL

A more realistic measurement interval in a "tactical" situation would be on the order of 30-60 seconds between measurements. Therefore, the BZA was tested as in the previous scenario and average errors are depicted in Figure 7 and Figure 8 (page 31) for 30 second interval and Figure 9 and Figure 10 (page 32) for the 60 second interval. In both cases, the steep slope is evident at about one third of total measurements with

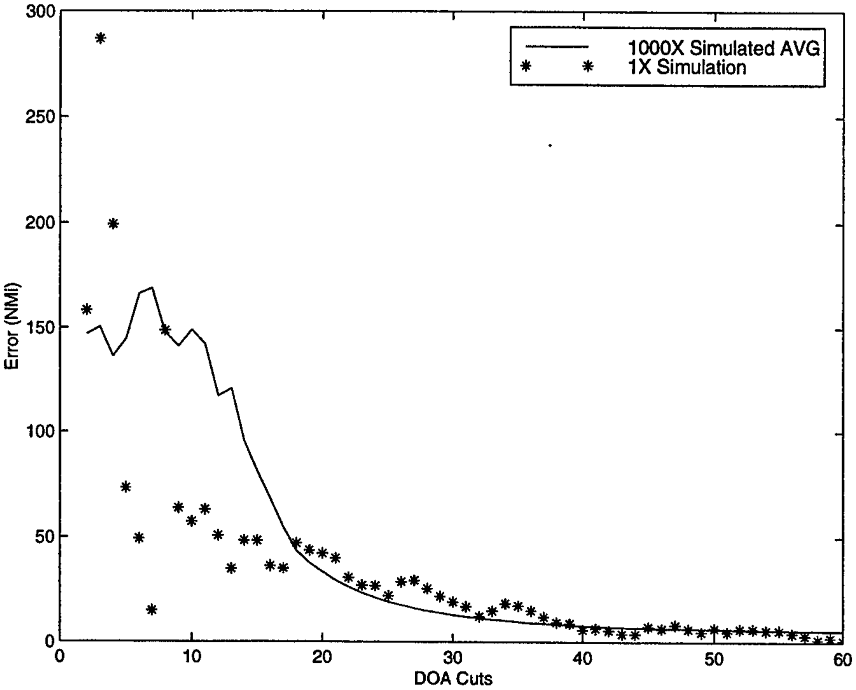

Figure 7: Averaged Error 30 Second Intervals 1000 Loops

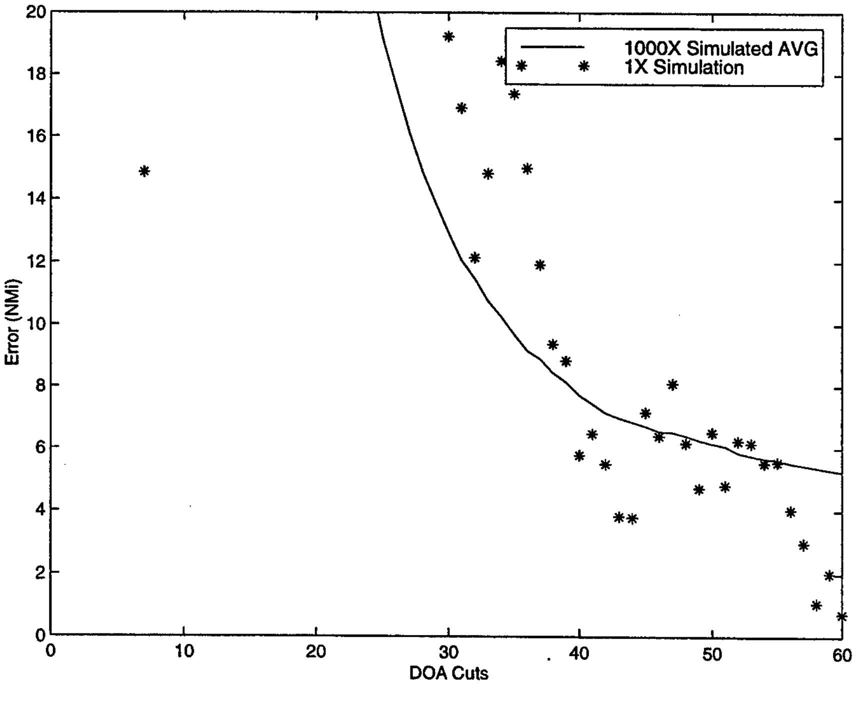

Figure 8: Averaged Error(<20 Nmi) 30 Second Intervals 1000 Loops

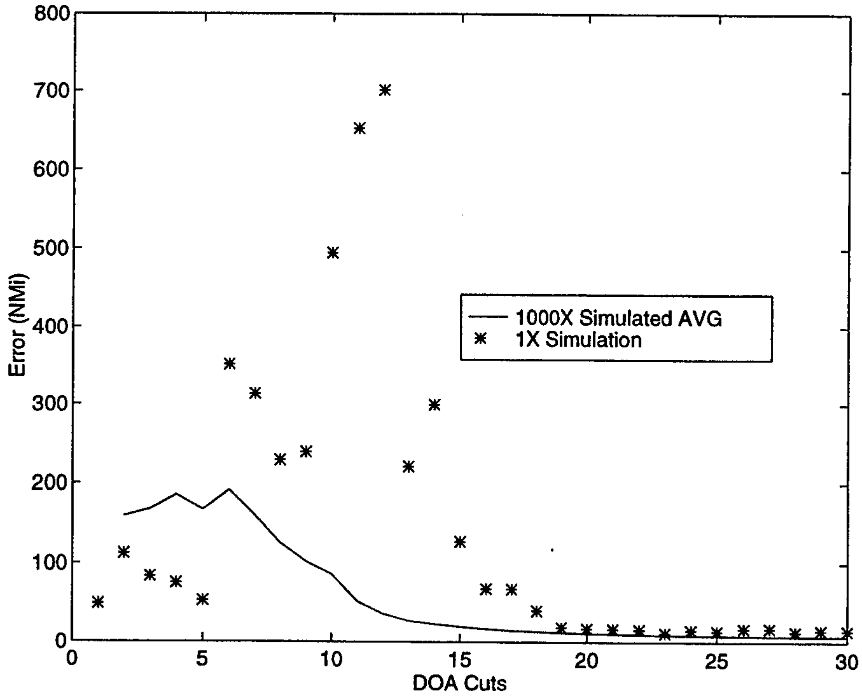

Figure 9: Averaged Error 60 Second Intervals 1000 Loops

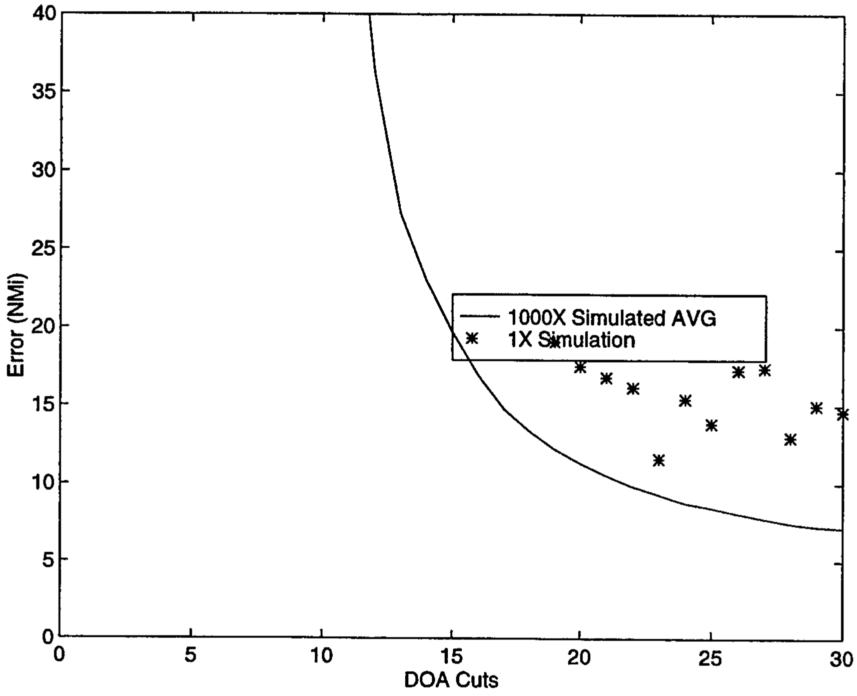

Figure 10: Averaged Error (<20 Nmi) 60 Second Intervals 1000 Loops

a leveling off following the passage of the abeam position. Accuracy at the completion of the scenario was under 6 Nmi. for the 30 second interval and about 7 Nmi. for the 60 second interval. Obviously, more data results in more accuracy, but in the three cases (Basic, 30 sec., 60 sec.) presented, best accuracy is achieved just after 45 degrees of relative change in DOA.

D. EFFECTS OF START/STOP POSITION

In an effort to determine if the algorithm was affected by changes in position for start and stop of the measurements, the basic scenario was modified so that sampling interval remained 12 seconds, but the initial and end points were modified. Figures 11 through 13 (pages 34-35) show the effects of starting at 2 degrees S. Lat. And continuing North to the Equator for a total of 120 measurements. .Figures 14 through 16 (pages 35-36) depict the effects of starting at the Equator and continuing North to 2 degrees N. Lat., also for 120 measurements. The algorithm gave similar final results in both cases, for an average error at the conclusion of the scenario of about 6-7 Nmi. There is, however, a striking difference in the slope of the range error curves, the range error for the Equator start is much steeper and approaches near steady state by the thirtieth measurement. In contrast, the Equator finish scenario (Figure 12) requires over 60 measurements to achieve the same average accuracy. Clearly the preferred method to locate the target if the 90 degree option is not available is to start with the target at least abeam the aircraft and to continue on a course perpendicular to the radius of an arc described by the theoretical range to the target. This is due in part to the more rapid change in apparent

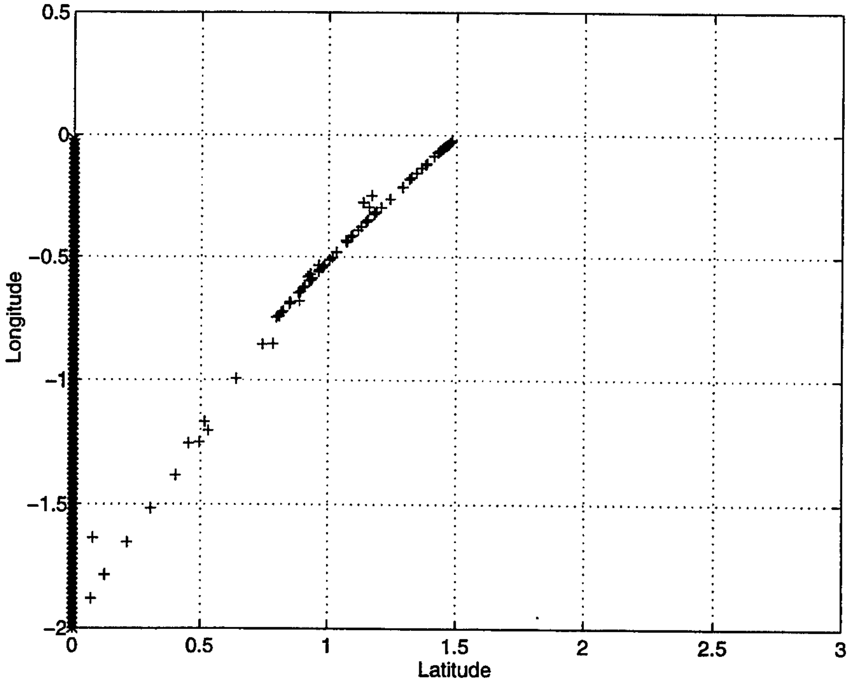

Figure 11: 2 Deg S. - 0 Deg Single Run Estimates

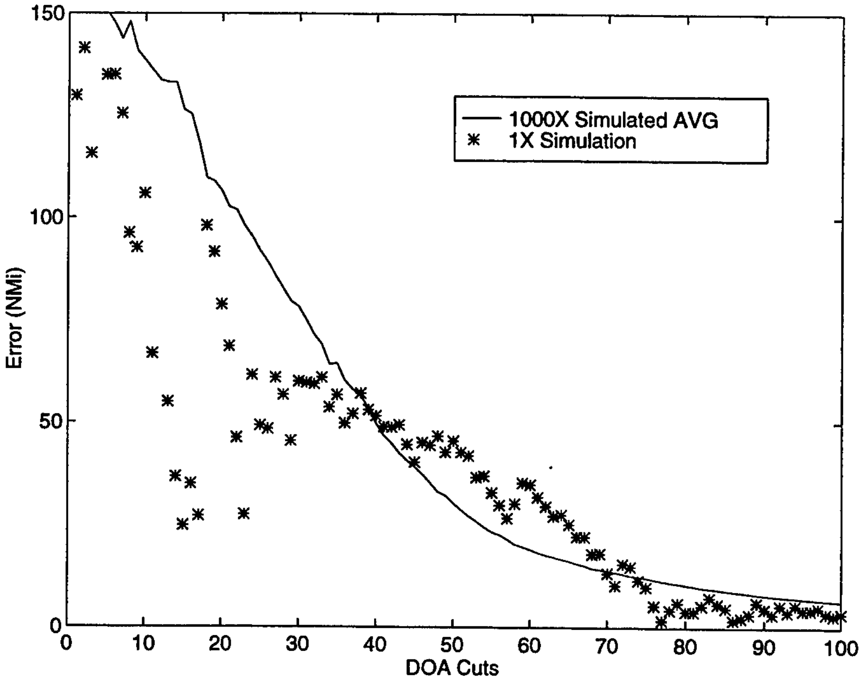

Figure 12: 2 Deg S - 0 Deg Averaged Error 1000 Loops

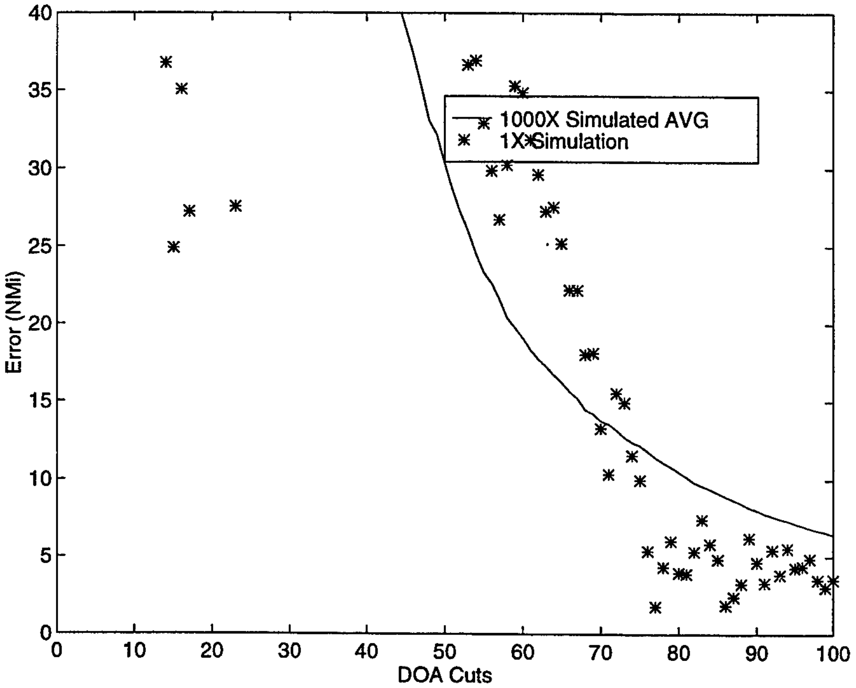

Figure 13: 2 Deg S - 0 Deg Averaged Error (<40 NMi)1000 Loops

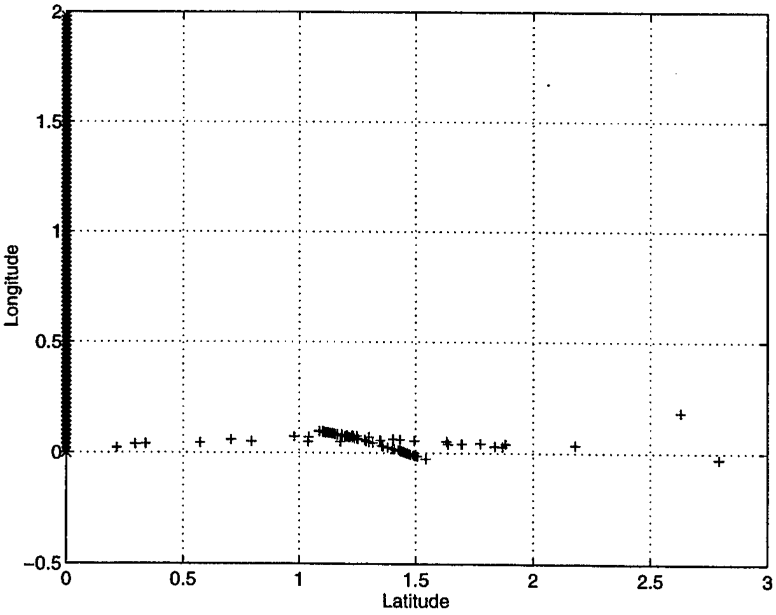

Figure 14: 0 Deg - 2 Deg N. Single Run Estimates

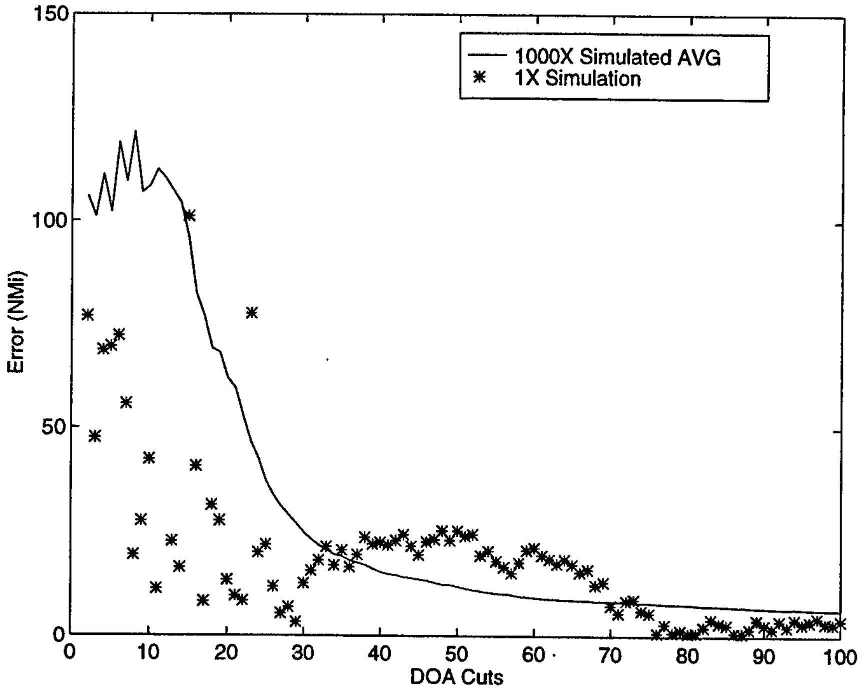

Figure 15: 0 Deg - 2 Deg N. Averaged Error 1000 Loops

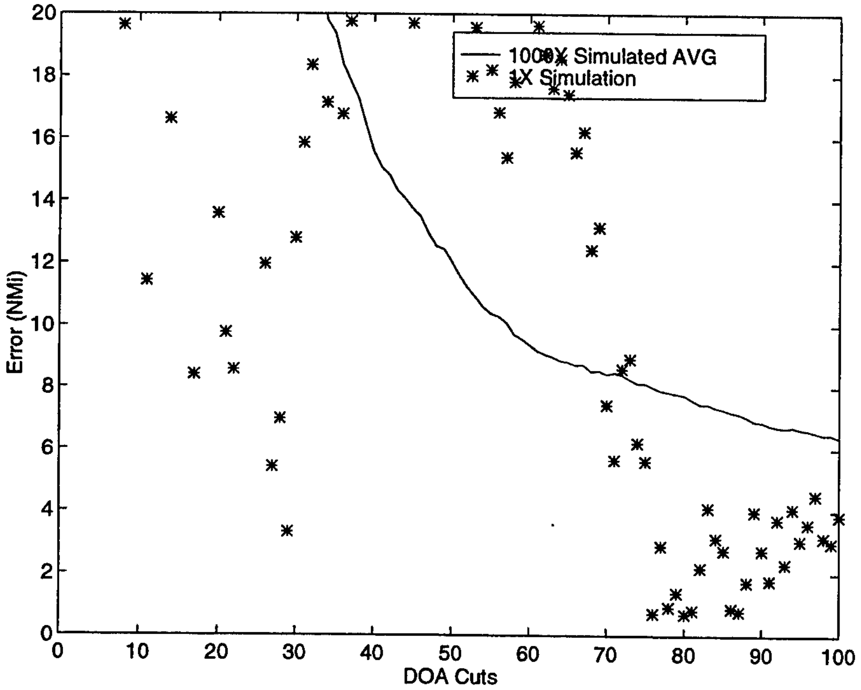

Figure 16: 0 Deg - 2 Deg N. Averaged Error (<20 Nmi) 1000 Loops

bearing from the aircraft to the target when the target is at the abeam position. Figures 11 and 14 show the "memory" of the algorithm where it becomes apparent that the algorithm relies heavily upon the first DOA as all estimates seem to come at or near the first relative bearing line.

VII. KALMAN FILTER RESULTS

A. KALMAN FILTER

The Kalman Filters tested for this thesis are representations of those developed and already cited by Mills and Coburn. Coburn's thesis relied on an inverse matrix manipulation that is not supported by the AYK-14. His example is presented as a comparison to the actual tested example, as presented by Mills. Mills developed a scalar approximation to the inverse of the P matrix. The same baseline scenario was run with both algorithms an yielded the results presented in Figure 17.

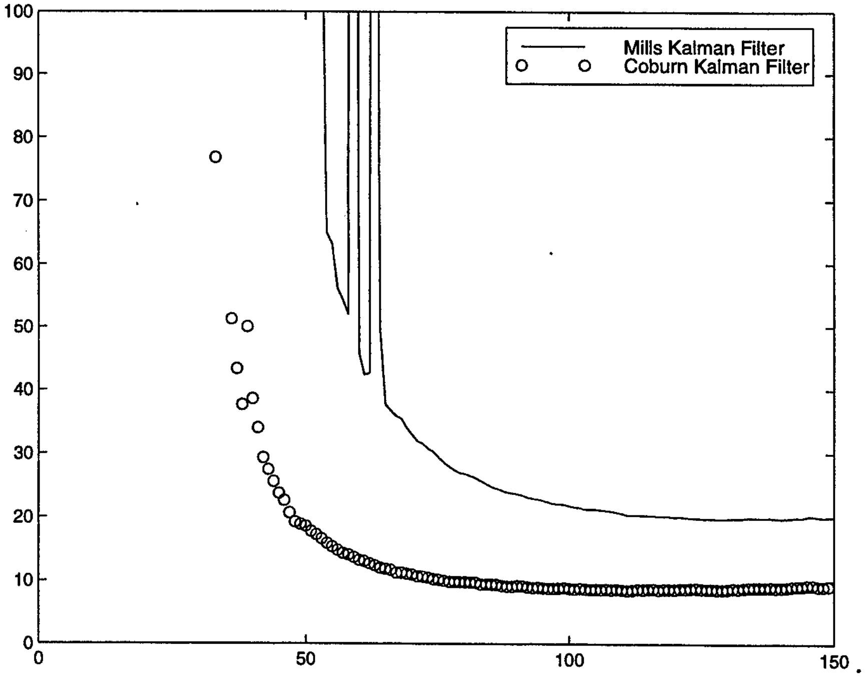

Figure 17: Comparison Between Mills and Coburn Kalman Filter

Thus, although Coburn's case is a clear winner, we cannot use his example since it cannot be supported by the current computer utilized by the typical ES aircraft.

B. BASIC SCENARIO

Figure 18, depicts a typical ES track along the Prime Meridian and the cluster of estimated positions for the target emitter for one run of the simulation. Accuracy was not as impressive as the BZA and is shown in Figure 19 for both one simulation run, and the average of 1000 simulations. The algorithm utilizes 115 floating point operations per

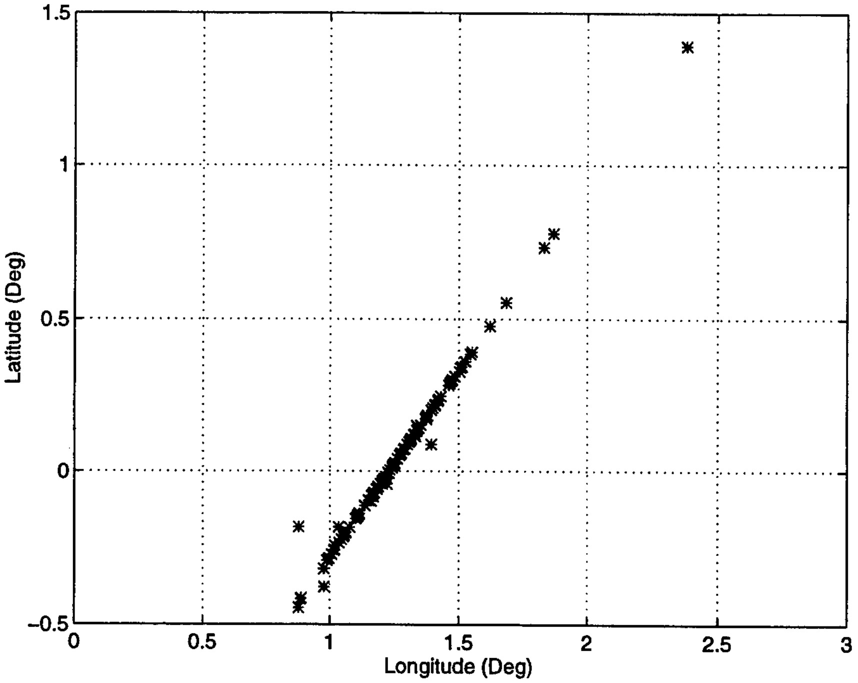

Figure 18: Single Run Kalman Filter Baseline Simulation.

measurement which is almost a 25 percent reduction in necessary calculations when compared to BZA. Finally, a simple time analysis was accomplished by evaluating the time to complete one thousand iterations of the complete scenario. The one thousand simulations took 1657.2 seconds to complete for an average of 1.6572 seconds for each loop through the scenario. Thus, with 150 measurements taken, the Kalman Filter (KF) process averages 0.011 seconds per measurement to arrive at a position estimate, which is almost 34 percent faster than the BZA. Figure 19 depicts the average error associated with the BZA under the conditions set in Chapter V. Figure 20 (shown on the next page) is a blown up depiction of average error focusing upon the area of convergence for the targeted emitter.

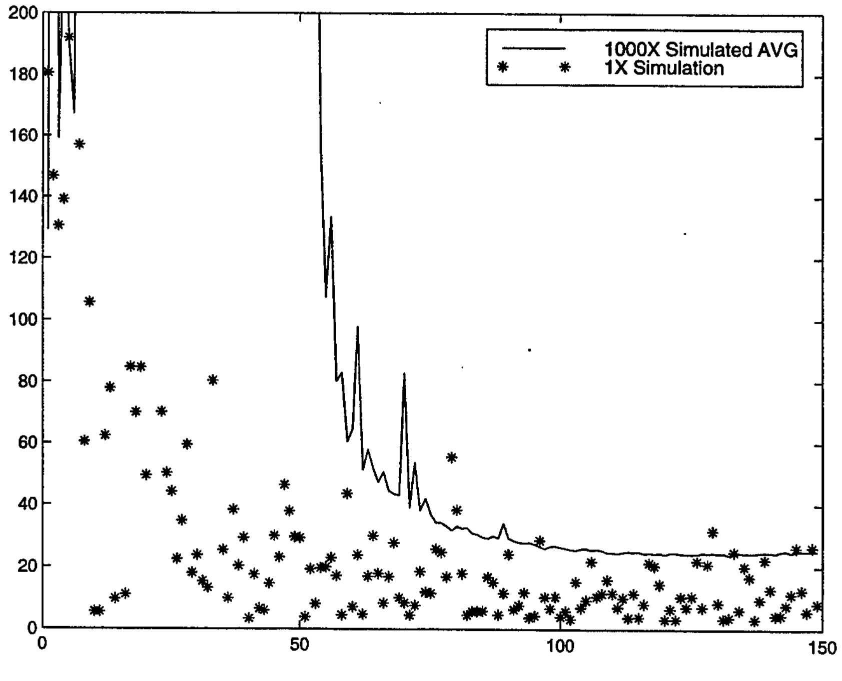

Figure 19: Averaged Error 1000 Simulations

The downward slope of the average error is very steep, starting at about 50 measurements and flattens out after 75 measurements, which corresponds with the aircraft passing the abeam position to the target emitter. Measurements were terminated at 150, which corresponds to a theoretical 90 degree change in target position. Final average range error is around 25 Nmi. at the termination of the test.

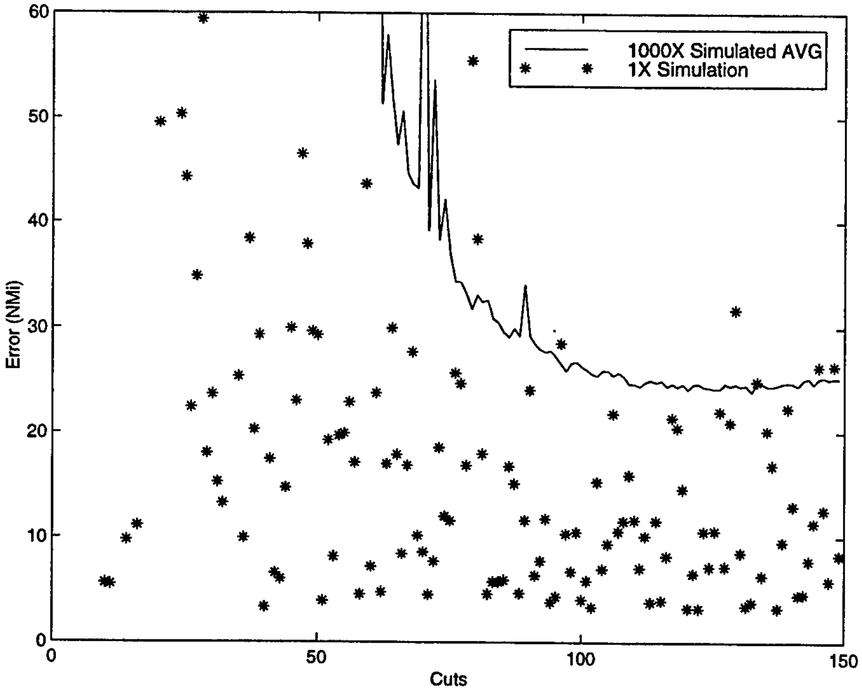

Figure 20: Averaged Error (<60 Nmi) 1000 Simulations

C. EFFECTS OF SAMPLING INTERVAL

As in Chapter VI, a more realistic measurement interval in a "tactical" situation would be on the order of 30-60 seconds between measurements. Therefore, the Kalman

Filter was tested as in the previous scenario and average errors are depicted in Figure 21 and Figure 22 (page 44) for 30 second interval and Figure 23 (page 44) and Figure 24 (page 45) for the 60 second interval. In both cases, the steep slope is evident at about one third of total measurements with a leveling off following the passage of the abeam position. There is a spike in both scenarios at around the abeam position, but in both cases the algorithm rapidly returns toward convergence on a minimum error. Accuracy at the completion of the scenario was about 26 Nmi. for the 30 second interval and about 28 Nmi. for the 60 second interval. In the case of the Kalman Filter it appears, more data results in more accuracy, but not a remarkable improvement when compared to the baseline. Again, in all three cases (Basic, 30 sec., 60 sec.) presented, best accuracy is achieved just after 45 degrees of relative change in DOA.

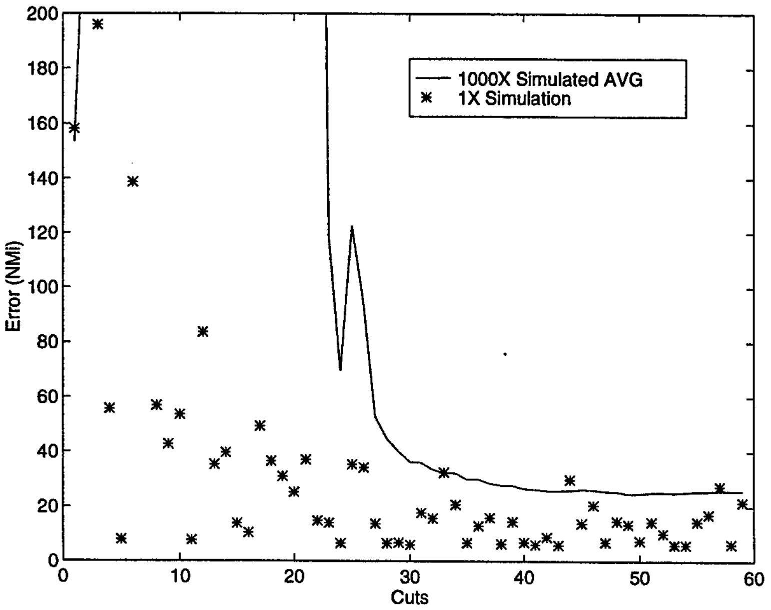

Figure 21: Averaged Error 30 Second Intervals 1000 Loops

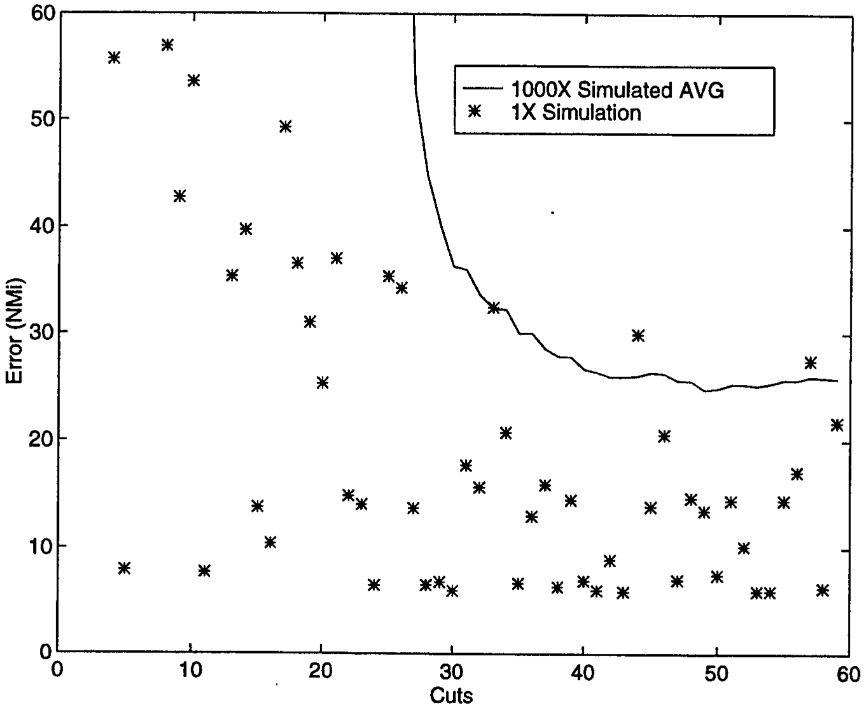

Figure 22: Averaged Error (<60 Nmi) 30 Second Intervals 1000 Loops

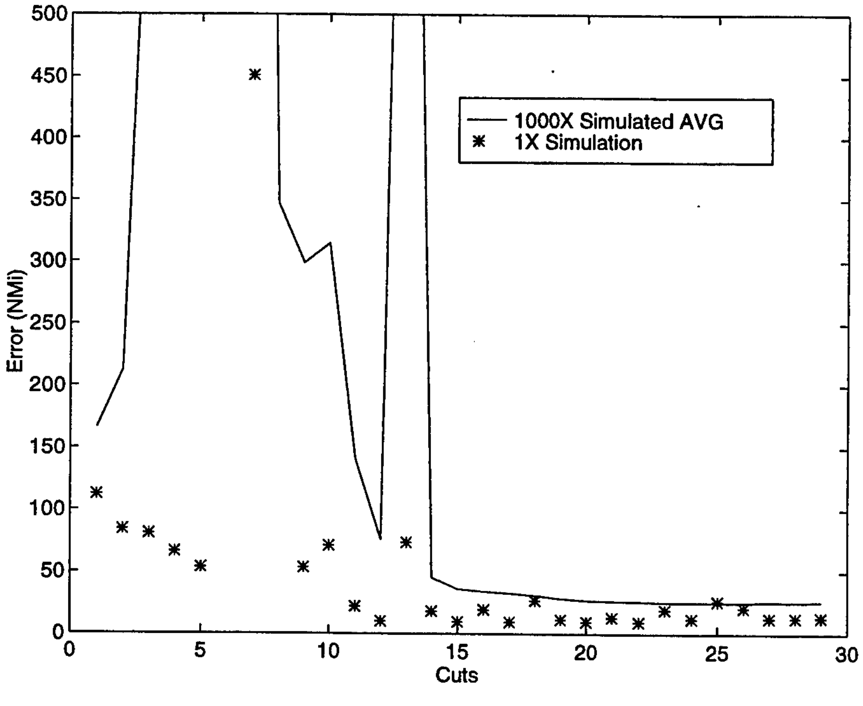

Figure 23: Averaged Error 60 Second Intervals 1000 Loops

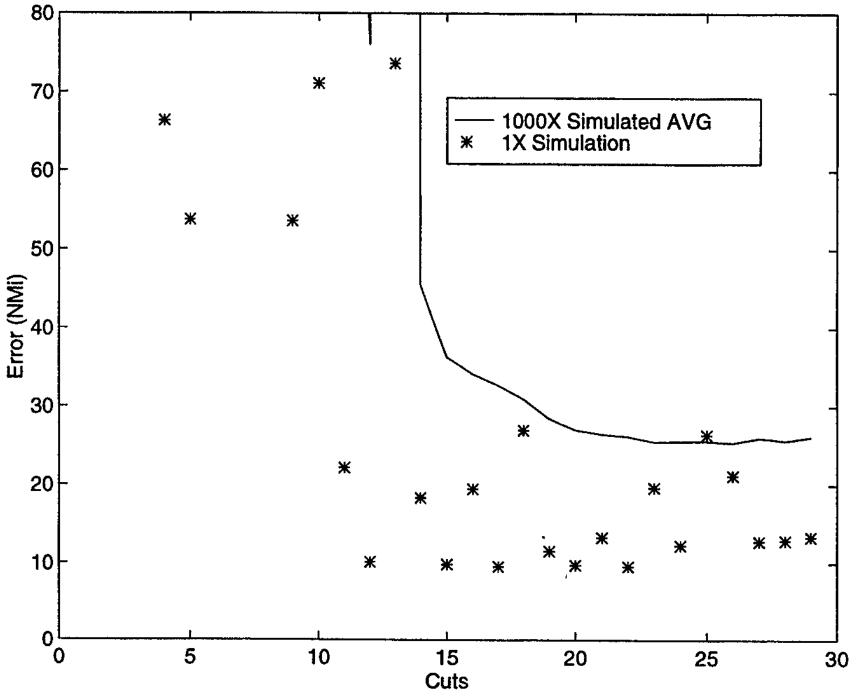

Figure 24: Averaged Error (<80 Nmi) 60 Second Intervals 1000 Loops

D. EFFECTS OF START/STOP POSITION

In an effort to determine if the Kalman Filter algorithm is affected by changes in position for start and stop of the measurements, the basic scenario was modified so that sampling interval remained 12 seconds, but the initial and end points were modified. Figures 25 through 27 (pages 45-46) show the effects of starting at 2 degrees S. Lat. And continuing North to the Equator for a total of 100 measurements. Figures 28 through 30 (pages 47-48) depict the effects of starting at the Equator and continuing North to 2 degrees N. Lat., also for 100 measurements. Both algorithms gave similar final results for an average error at the conclusion of the scenario of about 35 Nmi. There is, however, a striking difference in the slope of the range error curves, the range error for the Equator start is much steeper and approaches near steady state by the fiftieth measurement. In contrast, the Equator finish scenario (Figure 26) requires over 80 measurements to achieve the same average accuracy. Clearly the preferred method to locate the target, if the 90 degree option is not available, is to start with the target abeam the aircraft and to continue on a course perpendicular to the radius of an arc described by the theoretical range to the target. This is in agreement with results obtained utilizing BZA.

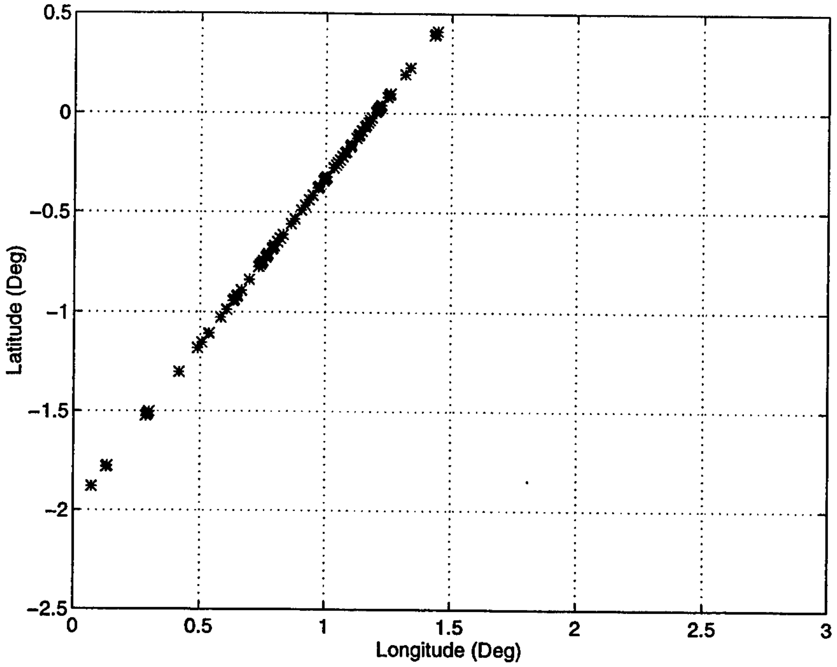

Figure 25: 2 Deg S. - 0 Deg Single Run Estimates

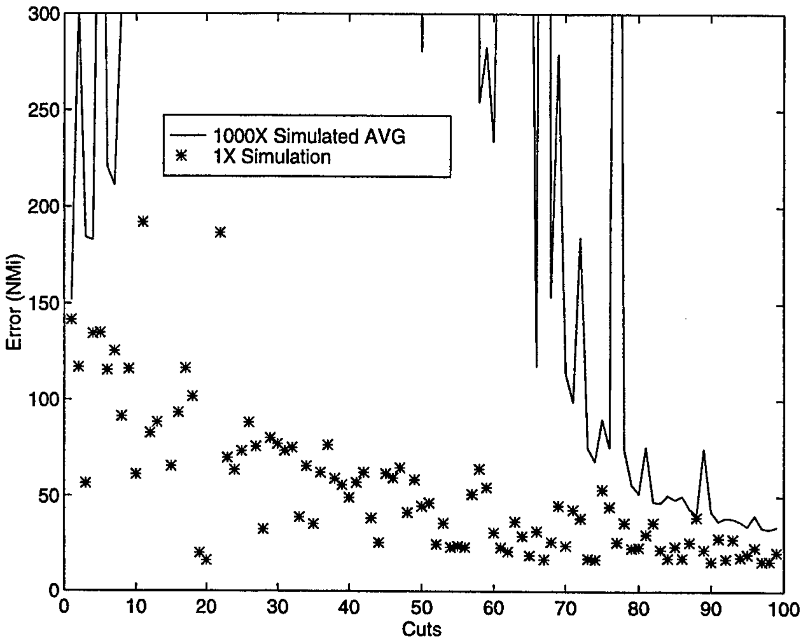

Figure 26: 2 Deg S - 0 Deg Averaged Error 1000 Loops

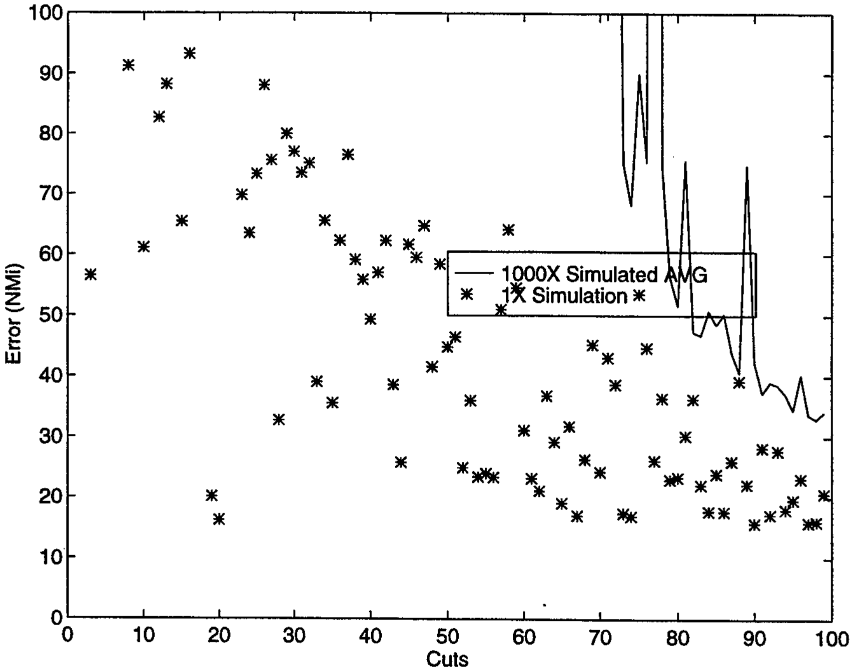

Figure 27: 2 Deg S - 0 Deg Averaged Error (<100 NMi)1000 Loops

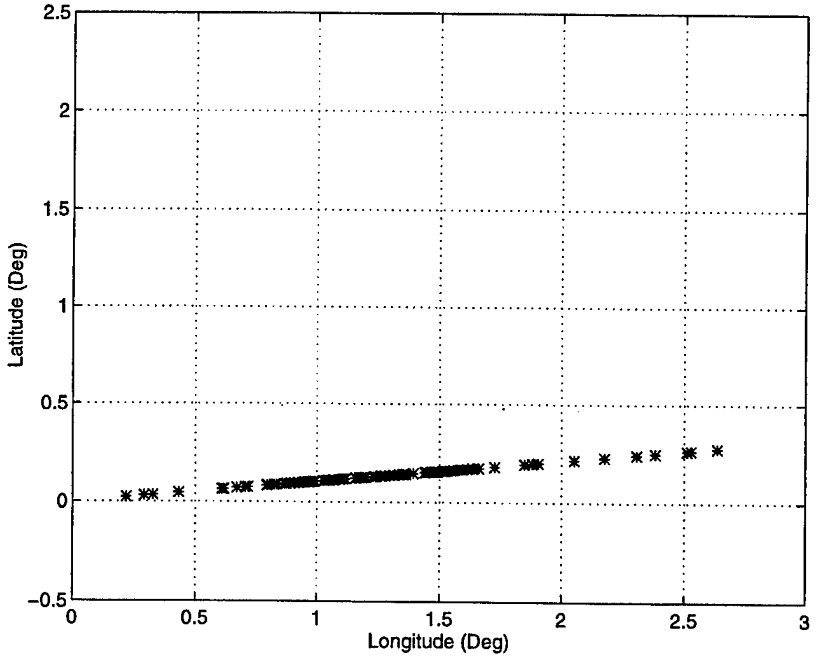

Figure 28: 0 Deg - 2 Deg N. Single Run Estimates

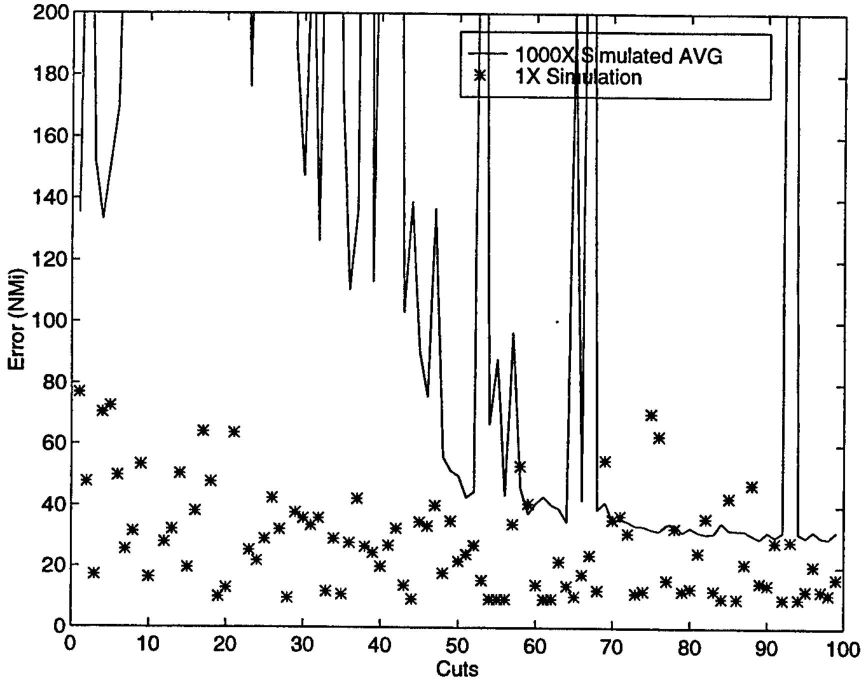

Figure 29: 0 Deg - 2 Deg N. Averaged Error 1000 Loops

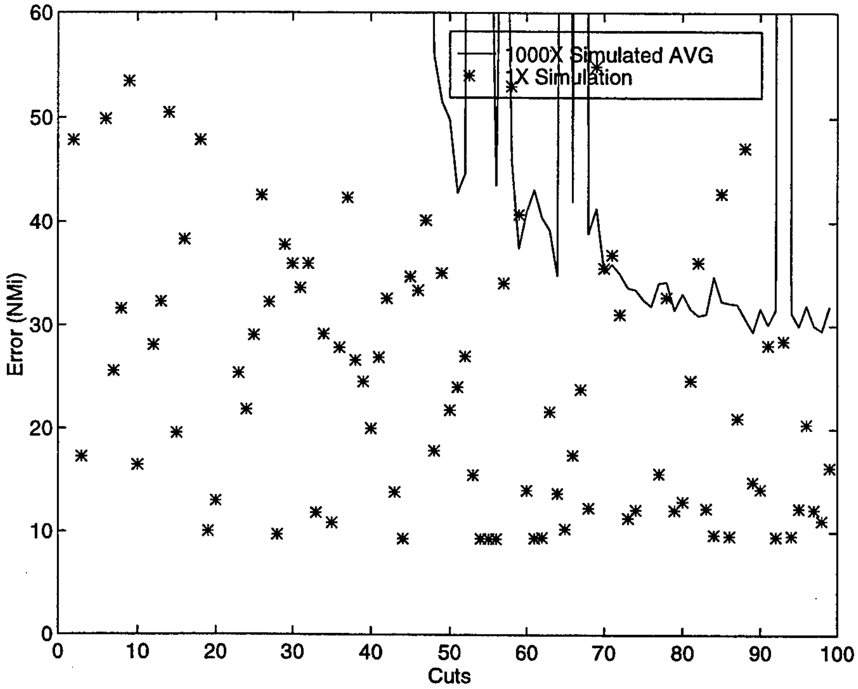

Figure 30: 0 Deg - 2 Deg N. Averaged Error (<60 Nmi) 1000 Loops

Figures 25 and 28 show the "memory" of the algorithm where it becomes apparent that the algorithm relies heavily upon the first DOA as all estimates seem to come at or near the first relative bearing line. This is expected with the Kalman Filter as the first DOA is continuously smoothed as the algorithm proceeds, and is consistent with BZA.

VIII. EXTENDED KALMAN FILTER RESULTS

A. THE EXTENDED KALMAN FILTER

The extended Kalman Filter was programmed to mirror the work of Coburn and Mills, who both relied upon a more accurate measuring device then the one simulated for this scenario. Analysis of the algorithm was incomplete due to the volatility of the results.

B. BASELINE SCENARIO

The results of the baseline scenario were so poor that an attempt was made to see if the problem was numerically unstable due to the use of latitude and longitude. The use of the X-Y plane could facilitate solving the equations and a conversion based on aircraft present position and heading could be used to arrive at a solution. Unfortunately, for DF accuracy greater than 1 degree, the filter always diverged. An attempt was made to provide only exact initialization data to start the algorithm, with like results. Thus, there is no apparent utility in this method at this time.

IX. CONCLUSIONS

Although the stated objective of this thesis was to determine if the Kalman Filter algorithms would be faster and more efficient than the current implementation, it is noted that accuracy is the ultimate goal and should not be sacrificed in order to increase speed, and efficiency.

A. EFFICIENCY

Clearly, the evidence presented in chapters VII and VIII shows that the Kalman Filters are faster and require fewer floating point operations. The AN/AYK-14 is a time share computer that has to be optimally programmed to expedite all tasks; however, accuracy remains paramount.

B. MEMORY REQUIREMENTS

Again, the evidence shows the Kalman Filter requires almost 25 percent fewer memory registers in order to complete the necessary equations to effectively locate target emitters. On the other hand, the number of required memory registers involved are insubstantial, and the case can be made that this is not an important criteria for determining which algorithm is best utilized.

C. ACCURACY

Without doubt, the deciding factor for aircraft operators given the limited disparity in time and memory requirements is effective accuracy. The BZA is clearly far more accurate than any of the Kalman Filters. Thus, despite the increased processing demands placed upon the AN/AYK-14, the BZA should be retained. The BZA was more accurate in virtually every scenario encountered. The effects of geometry and sample interval were noticeble, but were still better than both Kalman Filters (Coburn and Mills) in the baseline scenario.

D. FUTURE DEVELOPMENTS

As the algorithms present themselves now, the BZA is the clear winner. With the advent of portable terminals that interface with aircraft systems, the lack of speed and computing power now experienced will be overcome by the processing power of microchips. It appears that for this situation at this time, the Kalman Filter estimators are not the answer. That is not to say a more complicated Kalman Filter used in conjunction with an Extended Kalman Filter could not be developed, but the BZA is more accurate and not remarkably slower than the currently programmed examples evaluated.

E. AREAS FOR FUTURE RESEARCH

As part of the comparison of the various algorithms, the scenarios and variables have been kept very simple, but true inflight experience has yielded results that are almost never as accurate as those presented in this paper. I would propose that the effect of utilizing the BZA with an 8-bit computing system inflight vice the 32-bit system utilized for this thesis be investigated. Also, I have kept the sample interval regular and not extreme in length. If real world constraints and limitations are considered, the results may see some impact. Additionally, I have not considered the various time averaging program steps the ALQ-99 applies to incoming DOAs when the time between reception is long.

The EA-6B is the planned C2W aircraft for the next twenty years, and as such, every effort must be made to optimize the ES capability of the crew as the threat advances in technology and processing power.

APPENDIX A. BOOZOO CODE

This appendix is a representative case of Opperman's code FORTRAN written in MATLAB.

```
%%%%%               BOOZOO.M               %%%%%
%%                                                %%
%                                                   %
%       Maj S.P. Jones, USMC
% This script is a MATLAB adaptation of the FORTRAN subroutine
% program developed by Daniel Opperman in June, 1997.

%       flong is aircraft position longitude
%       flat is aircraft position latitude
%       azz is the raw doa azimuth bearing measurement
%       wt is the weighting factor for the measurement
%       i is the index denoting the number of the present measurement
%       ncuts is the total number of measurements
%       elong is the estimated emitter longitude in radians
%       elat is the estimated emitter latitude in radians


clf
%%% Initiate the program
randn('seed',2)             % set seed generator

pirad = pi/180;             % conversion factor from deg-->rad
flong = 0;                  % 0 degrees long (rad)
flat = -1.5*pirad; % 1.5 deg S. (rad)
azzout = [];
flatout = [];
flongout = [];
elatmat = [];
elongmat = [];
itout = [];
tgtlatmat = [];
tgtlongmat = [];
range_error = [];
tgtlong = 1.5*pirad;        % 1.5 deg E. (rad)
tgtlat = 0;                 % at the equator
wt = 1;
wttable = [.8 .96 .87 .74 .51 .5 .47 .46 .48 .57 .57 ...
        .62 .42 .64 .61 .64 .62 .67 .67 .65 .6 .62 ...
        .63 .7 .39 .58 .51 .45 .47 .47 .57 .78 .88 ...
        .88 .99 .78 .47];
index = [0:149];

for i = 1:150

dferror = 10*pirad*(randn);
```

```
%azz1 = atan2((tgtlat-flat),(tgtlong-flong))
azz = (pi/2)-atan((tgtlat-flat)/(tgtlong-flong)) + dferror;
azzout = [azzout,azz];

%%% DETERMINE WEIGHT

if azz < 180

        wtindex = round((azz + 2.5)/5);

else

        wtindex = round((360 - azz + 2.5)/5);

end                 %%% if azz<180

wt = wttable(wtindex);

if i==1
        test = 2;
        %
        %% INITIALIZE A AND B MATRICES IF THIS IS FIRST MEASUREMENT%%

        a11 = 0;
        a12 = 0;
        a13 = 0;
        a22 = 0;
        a23 = 0;
        a33 = 0;

        b11 = 0;
        b12 = 0;
        b13 = 0;
        b22 = 0;
        b23 = 0;
        b33 = 0;

end                 %end if i==1

%%      CALCULATE DIRECTIONAL NUMBERS XS,YS,ZS for Ith CUTTING PLANE %%

slat = sin(flat);
clat = cos(flat);
slong = sin(flong);
clong = cos(flong);
sazz = sin(azz);
cazz = cos(azz);

XS = (slong*cazz - slat*clong*sazz) * wt;
YS = (-clong*cazz - slat*slong*sazz) * wt;
ZS = clat*sazz*wt;

%%% CALCULATE THE A MATRIX

a11 = a11 + XS^2;
```

```
a12 = a12 + XS*YS;
a13 = a13 + XS*ZS;
a22 = a22 + YS^2;
a23 = a23 + YS*ZS;
a33 = a33 + ZS^2;

% Determine the vector components X,Y,Z of A/C position vector

X = clong*clat;
Y = slong *clat;
Z = slat;

% Calculate the B matrix

b11 = b11 + Y^2 + Z^2;
b12 = b12 - X*Y;
b13 = b13 - X*Z;
b22 = b22 + X^2 + Z^2;
b23 = b23 - Y*Z;
b33 = b33 + X^2 + Y^2;

%%%% Calculate the Co-Factors of the A Matrix

c11 = a22*a33 - a23^2;
c12 = a13*a23 - a12*a33;
c13 = a12*a23 - a13*a22;
c22 = a11*a33 - a13^2;
c23 = a12*a13 - a11*a23;
c33 = a11*a22 - a12^2;

u = c11;
v = c12;
w = c13;

elong = 999;
elat = 999;

if i < 2   %%% Need at least two cuts
          elat = 0;
          elong = 0;

end                  % If i<2
iterations = 0;
if i == 150
          flops(0)

end

while test > 1.00000000    %%%Iterative Loop

          iterations = iterations +1;


          elon1 = elong;
          elat1 = elat;
```

```
        A1 = b11*u + b12*v + b13*w;
        A2 = b12*u + b22*v + b23*w;
        A3 = b13*u + b23*v + b33*w;

%%%  DETERMINE COMPONENTS OF POSITION VECTOR

        u = A1*c11 + A2*c12 + A3*c13;
        v = A1*c12 + A2*c22 + A3*c23;
        w = A1*c13 + A2*c23 + A3*c33;

%%% DETERMINE LAT AND LONG OF TGT EMITTER

        elong = atan(v/u);
        elat = atan(w/sqrt(u^2 + v^2));
        test = abs(elon1-elong) + abs(elat1-elat);


end      %%%(while test >1e-8)
if i ==150

flops

end

itout = [itout,iterations];

test = 2;

flatout = [flatout,flat];
flongout = [flongout,0];
flat = flat + ((6*12/60*1/60)*pirad);

tgtlatmat = [tgtlatmat,tgtlat/pirad];
tgtlongmat = [tgtlongmat,tgtlong/pirad];

elatout = elat/pirad;
elatmat = [elatmat,elatout];

elongout = elong/pirad;
elongmat = [elongmat,elongout];

error = sqrt((elat-tgtlat)^2 + (elong - tgtlong)^2)/pirad*60;

range_error = [range_error,error];

end      %%% for i = 1 : 360

figure(1)
plot(flongout/pirad,flatout/pirad,'b*',elongmat,elatmat,'r+',...
        tgtlongmat,flatout/pirad,'y-',index,tgtlatmat,'y-')
xlabel('Longitude (deg) East'),ylabel('Latitude (deg)')
axis([0,3,-1.5,1.5])
print -deps fig4ch
```

```
figure(2)

plot(range_error)
title('distance error (Nmi)')
print -deps bz3fig2

figure(3)
plot(elatmat,'+')
title('Predicted Target Latitude (deg) Target Located at 0 deg')
hold
plot(tgtlatmat)
axis([0,i,-5,5])

figure(4)
plot(elongmat,'+')
title('Predicted Target Longitude (deg) Target Located at 1.5 deg')
hold
plot(tgtlongmat)
axis([0,i,-5,5])

figure(5)
plot(itout)
title('Iterations Required to Make Estimate')
axis([0,i,0,max(itout)+2])
```

APPENDIX B. COBURN KALMAN FILTER CODE

This appendix is a representative case of Coburn's FORTRAN code written in MATLAB.

```
%%%%%               CKFAVG.M               %%%%%
%%                                                 %%
%                                                    %
%       Maj S.P. Jones, USMC
% This function is a MATLAB adaptation of the FORTRAN
% program developed by L. Laddie Coburn in June, 1972.

%       flong is aircraft position longitude
%       flat is aircraft position latitude
%       vel is the aircraft velocity
%       azz is the raw doa azimuth bearing measurement
%       wt is the weighting factor for the measurement(not used)
%       i is the index denoting the number of the present measurement
%       ncuts is the total number of measurements
%       toa is time of arrival
%       elong is the estimated emitter longitude in radians
%       elat is the estimated emitter latitude in radians
%       acc is df accuracy 1 sigma usually 10 degrees


clear all

%%% Initiate the program
randn('seed',2)             % set seed generator
ncuts = 150;
index = [1:ncuts-1];
degperrad = 180/pi;         % conversion factor from deg-->rad

elatmatc = zeros(1,ncuts-1);
elongmatc = zeros(1,ncuts-1);
rngerrsumc = zeros(1,ncuts-1);

total = 10;

for loops = 1:total

flatout = [];
flongout = [];
azzout = [];
elatoutc = [];
elongoutc = [];
rngeroroutc = [];
```

```
flong1 = 0/degperrad;                    % 0 degrees long (rad)
flat1 = -1.5/degperrad;      % 1.5 deg S. (rad)
flong = flong1;
flat = flat1;

tgtlong = 1.5/degperrad;   % 1.5 deg E. (rad)
tgtlat = 0/degperrad;                    % at the equator
tgtlatout = tgtlat/degperrad*ones(1,ncuts-1);
tgtlongout =1.5*ones(1,ncuts-1);

acc = 10/degperrad;                      % df accuracy is about +/- 10 degrees
vel = 360;                               % A/C velocity is 360 knots
rngest = 1.414*90;                               % initial range estimate always 150
toa = 0;

for i = 1:ncuts      %travel 180 NMi measuring every 12 sec

dferror = acc*(randn);

azz = (pi/2)-atan((tgtlat-flat)/(tgtlong-flong)) + dferror;

azzout = [azzout,azz];
azzdeg = azz*degperrad;

if i == 1

        p11 = 10000;                % P matrix
        p12 = 0;
        p21 = 0;
        p22 = 10000;
        q11 = 0;             % Q matrix
        q12 = 0;
        q22 = 0;
        oldd11 = 1;                 % D
        oldd12 = 0;
        oldd21 = 0;
        oldd22 = 1;
        w11 = 1;
        w12 = 0;
        w21 = 0;
        w22 = 1;
        R = (acc)^2;                % (10 degree std dev)^2 = var

        g1 = p11/(p11+R);           % G
        g2 = p12/(p11+R);


        thtd = azzdeg;              % thtd will be the filtered df cut
        thtd1 = azzdeg;             % thtd1 will the smoothed first cut

        thetadot = vel/rngest*sin(azz)*degperrad/3600;        % theta dot deg/sec
        thetadot1 = thetadot;               % smoothed first theta dot
        toa = 0;
        time1 = toa;                % dummy variable for computing elpased time
        t = 0;                      % elapsed time between cuts
        k2 = 0;
        k3 = 1;

else

        t = toa - time1;
        time1 = toa;                        %reset time1 for next cut
        tt = t/1000;
        q11 = tt;
        q12 = tt^(1.5);
        q22 = tt^2;


%%% KALMAN RECURSION EQUATIONS %%%%

newp11 = p11*(1-g1) + 2*p12*t - (p12*g1 + p11*g2)*t + ...
         (p22 - p12*g2)*t^2 + q11;

newp12 = p12*(1-g1) + (p22 - p12*g2)*t + q12;

newp22 = p22 - p12*g2 + q22;

g1 = newp11/(newp11 + R);
g2 = newp12/(newp11 + R);

tptd = thtd + thetadot*t;

e = azzdeg - tptd;

thtd = tptd + g1*e;                         % filtered df cut
thetadot = thetadot + g2*e;         % bearing rate


%%% SMOOTHING EQUATIONS FOR FIRST CUT %%%%
%%   Coburn relied on inverse
%%  matrix fumction that is not supported by CMS-2.

P =     [p11    p12
         p21    p22];

Pinv = inv(P);
pin11 = Pinv(1,1);
pin12 = Pinv(1,2);
pin22 = Pinv(2,2);

qp11 = q11*pin11 + q12*pin12;
qp12 = q11*pin12 + q12*pin22;
qp21 = q12*pin11 + q22*pin12;
qp22 = q12*pin12 + q22*pin22;

d11 = oldd11*(1-qp11) - oldd12*qp21 + oldd11*qp21*t;
d12 = oldd12*(1-qp22) - oldd11*qp12 + oldd11*(qp22-1)*t;
d21 = oldd21*(1-qp11) - oldd22*qp21 + oldd21*qp21*t;
d22 = oldd22*(1-qp22) - oldd21*qp12 + oldd21*(qp22-1)*t;
```

```
olddl1 = d11;
oldd12 = d12;
oldd21 = d21;
oldd22 = d22;

thtd1 = thtd1 + (d11*g1 + d12*g2)*e;
thetadot1 = thetadot1 + (d21*g1 + d22*g2)*e;

p11 = newp11;
p12 = newp12;
p22 = newp22;

thti = thtd/degperrad;
tht1i = thtd1/degperrad;

tat = tan(tht1i) - tan (thti);
wav = (flat1 + flat)/2;
tila = ((flong-flong1)*cos(wav) + flat1*tan(tht1i) - flat*tan(thti))/tat;
tla = ((flong-flong1)*cos(tila) + flat1*tan(tht1i) - flat*tan(thti))/tat;

elat = tla*degperrad;

tlo = flong + (tla - flat)*tan(thti)/cos(tila);
elong = tlo*degperrad;


elatoutc = [elatoutc,elat];
elongoutc = [elongoutc,elong];

rngerror = sqrt((tgtlat-tla)^2 + (tgtlong-tlo)^2)*degperrad*60;
rngeroroutc = [rngeroroutc,rngerror];

end      %(if i==1

oldlat = flat;
oldlong = flong;
toa = toa + 12;
flat = flat + 12/60*6/60/degperrad;
flatout = [flatout,flat];

end %  for i = 1:91

elatmatc = elatmatc + elatoutc;
elongmatc = elongmatc + elongoutc;
rngerrsumc = rngerrsumc + rngeroroutc;

end %  for loops = 1:

elatavgc = elatmatc/total;
elongavgc = elongmatc/total;
rngerravgc = rngerrsumc/total;

figure(1)
plot(index,elatavgc,'*b',index,tgtlatout,'r')
title('Estimated Latitude')
```

```
axis([0,ncuts,-5,5])

figure(2)
plot(index,elongavgc,'*b',index,tgtlongout,'r')
title('Estimated Longitude')
axis([0,ncuts,0,3])

figure(3)
plot(rngerravgc)
title('Error (Nmi)')
axis([0,ncuts,0,100])
```

APPENDIX C. MILLS' KALMAN FILTER CODE

This appendix is a representative case of Mills' FORTRAN code written in MATLAB.

```
%%%%%                MKFAVG.M                 %%%%%
%%                                                      %%
%                                                          %
%       Maj S.P. Jones, USMC
% This function is a MATLAB adaptation of the FORTRAN
% program developed by E.H. Mills in March, 1973.

%       flong is aircraft position longitude
%       flat is aircraft position latitude
%       vel is the aircraft velocity
%       azz is the raw doa azimuth bearing measurement
%       i denotes the number of the present measurement
%       toa is time of arrival
%       elong is the estimated emitter longitude in radians
%       elat is the estimated emitter latitude in radians
%       acc is df accuracy 1 sigma usually 10 degrees
%     tht is angle theta  True Azimuth - True Heading I have
%     modeled the system around an aircraft heading 000 and
%       eliminated this calculation
%       tht1 is the smoothed first azimuth
%       thetadot is the bearing rate in rad/sec
%       thetadot1 is the smoothed first bearing rate in rad/sec

clear all

%%% Initiate the program

randn('seed',2)           % set seed generator
ncuts = 150;
index = [1:ncuts-1];
degperrad = 180/pi;       % conversion factor from deg-->rad

elatmat = zeros(1,ncuts-1);
elongmat = zeros(1,ncuts-1);
rngerrsum = zeros(1,ncuts-1);

total = 1;

for loops = 1:total

loops
flatout = [];
flongout = [];

azzout = [];
```

```
elatout = [];
elongout = [];
rngerrorout = [];

flong1 = 0/degperrad;                    % 0 degrees long (rad)
flat1 = -1.5/degperrad;      % 1.5 deg S. (rad)
flong = flong1;
flat = flat1;

tgtlong = 1.5/degperrad;   % 1.5 deg E. (rad)
tgtlat = 0/degperrad;                    % at the equator
tgtlatout = tgtlat/degperrad*ones(1,ncuts-1);
tgtlongout =1.5*ones(1,ncuts-1);

acc = 10/degperrad;                      % df accuracy is about +/- 10 degrees
vel = 360;                               % A/C velocity is 360 knots
rngest = 1.414*90;                              % initial range estimate always 150
toa = 0;

for i = 1:ncuts      %travel 180 NMi measuring every 12 sec

dferror = acc*(randn);

azz = (pi/2)-atan((tgtlat-flat)/(tgtlong-flong)) + dferror;

azzout = [azzout,azz];
azzdeg = azz*degperrad;

if i == 1

        p11 = 10000;                % P matrix
        p12 = 0;
        p22 = 10000;
        q11 = 0;            % Q matrix
        q12 = 0;
        q22 = 0;
        d11 = 1;            % D
        w11 = 1;
        w12 = 0;
        w21 = 0;
        w22 = 1;
        R = (acc)^2;                % (10 degree std dev)^2 = var

        g1 = p11/(p11+R);           % G
        g2 = p12/(p11+R);

        tht = azz;
        tht1 = azz;

        thetadot = vel/rngest*sin(azz)/3600;         % theta dot deg/sec
        thetadot1 = thetadot;                  % smoothed first theta dot
        toa = 0;
        time1 = toa;                % dummy variable for computing elpased time
        t = 0;                      % elapsed time between cuts
```

```
else

        t = toa - time1;
        time1 = toa;                    %reset time1 for next cut
        tt = t/1000;
        q11 = tt;
        q12 = tt^(1.5);
        q22 = tt^2;


%%% KALMAN RECURSION EQUATIONS %%%%

newp11 = p11*(1-g1) + 2*p12*t - (p12*g1 + p11*g2)*t + ...
        (p22 - p12*g2)*t^2 + q11;

newp12 = p12*(1-g1) + (p22 - p12*g2)*t + q12;

newp22 = p22 - p12*g2 + q22;

newg1 = newp11/(newp11 + R);
newg2 = newp12/(newp11 + R);

tpt = tht + thetadot*t;

e = azz - tpt;

tht = tpt + g1*e;                   % filtered df cut
thetadot = thetadot + g2*e;         % bearing rate


%%% SMOOTHING EQUATIONS FOR FIRST CUT %%%%
%%  These equations are extracted from Edward H. Mills thesis
%%  that builds upon Coburn's work.  Coburn relied on inverse
%%  matrix fumction that is not supported by CMS-2, and so I
%%  have used this group of equations as developed by Mills
%%  to better simulate operation onboard an actual aircraft.

smooth1 = 1 - p11*(1-g1)/R;
smooth2 = smooth1*t;
smooth3 = -p12 * (1-g1)/R;
smooth4 = 1 + smooth3*t;

neww11 = w11*smooth1 + w12*smooth2;
neww12 = w11*smooth3 + w12*smooth4;
neww21 = w21*smooth1 + w22*smooth2;
neww22 = w21*smooth3 + w22*smooth4;

w11 = neww11;
w12 = neww12;
w21 = neww21;
w22 = neww22;

d11 = d11 - (w11^2)*(p11 + R)/(R^2);

tht1 = tht1 + (w11/R)*e;
```

```
thetadot1 = thetadot1 + (w21/R)*e;

p11 = newp11;
p12 = newp12;
p22 = newp22;

g1 = newg1;
g2 = newg2;

thti = tht;
tht1i = tht1;

tat = tan(tht1i) - tan (thti);
wav = (flat1 + flat)/2;
tila = ((flong-flong1)*cos(wav) + flat1*tan(tht1i) - flat*tan(thti))/tat;
tla = ((flong-flong1)*cos(tila) + flat1*tan(tht1i) - flat*tan(thti))/tat;
tlo = flong + (tla - flat)*tan(thti)/cos(tila);

elat = tla*degperrad;

elong = tlo*degperrad;

elatout = [elatout,elat];
elongout = [elongout,elong];

rngerror = (sqrt((tgtlat-tla)^2 + (tgtlong-tlo)^2))*degperrad*60;
rngerrorout = [rngerrorout,rngerror];

end       %(if i==1

oldlat = flat;
oldlong = flong;
toa = toa + 12;
flat = flat + 12/60*6/60/degperrad;
flatout = [flatout,flat];

end %  for i = 1:ncuts

elatmat = elatmat + elatout;
elongmat = elongmat + elongout;
rngerrsum = rngerrsum + rngerrorout;

end %  for loops = 1:

elatavg = elatmat/total;
elongavg = elongmat/total;
rngerravg = rngerrsum/total;

figure(1)
plot(index,elatavg,'*b',index,tgtlatout,'r')
%title('Estimated Latitude 150 Cuts 10+/-Degree Accuracy Mills Kalman Filter')
xlabel('Cuts'), ylabel('Latitude (Deg)')
axis([0,ncuts,-5,5])
%print -deps mkf12f1
```

```
figure(2)
plot(index,elongavg,'*b',index,tgtlongout,'r')
%title('Estimated Longitude 150 Cuts 10+/-Degree Accuracy Mills Kalman Filter')
xlabel('Cuts'), ylabel('Longitude (Deg)')
axis([0,ncuts,0,3])
%print -deps mkf12f2

figure(3)
plot(index,rngerravg,'-',index,rngerrorout,'*')
legend('1000X Simulated AVG','1X Simulation')
%title('Error (NMi) 150 Cuts 10+/-Degree Accuracy Mills Kalman Filter')
xlabel('Cuts'),ylabel('Error (NMi)')
axis([0,ncuts,0,200])
%print -deps mkf12f3

figure(4)
plot(elongout,elatout,'*g')
ylabel('Latitude (Deg)'), xlabel('Longitude (Deg)')
grid
%title('Mills Kalman Filter Predictions 150 Cuts +/- 10 Degree Accuracy')
axis([0,3,-.5,1.5])
%print -deps mkf12f4

figure(5)
plot(index,rngerravg,'-',index,rngerrorout,'*')
legend('1000X Simulated AVG','1X Simulation')
%title('Error (NMi) 150 Cuts 10+/-Degree Accuracy Mills Kalman Filter')
axis([0,ncuts,0,60])
xlabel('Cuts'),ylabel('Error (NMi)')
%print -deps mkf12f5
```

LIST OF REFERENCES

1. Coburn, L. Laddie, Kalman Filtering Techniques Applied to Airborne Direction Finding and Emitter Location, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1972

2. Opperman, Daniel Charles, The Evaluation and Performance of a Least Squares Emitter Locating Algorithm, Master's Thesis, California State University, Northridge, May, 1977

3. Mills, Edward Harlan, Applying the Kalman Filter to the Emitter Location Problem Using Airborne Angle-of-Arrival Information, Master's Thesis, Naval Postgraduate School, Monterey, CA, March, 1973

4. Rauch, H.E., "Solutions to Linear Smoothing Problem," IEEE Transactions on Automatic Control, vol. AC-8, no. 4, pp.371-372, October 1963

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Road
   Quantico, VA 22134-5027

4. Director, Marine Corps Research Center 2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, VA 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC, Code C45
   3300 Russell Road
   Quantico, VA 22134-5130

6. Marine Corps Representative 1
   Naval Postgraduate School
   Code 037, Bldg 234, HA-220
   699 Dyer Road
   Monterey, CA 93940

7. Marine Corps Tactical Systems Support Activity 1
   Technical Advisory Branch
   Attn: Maj. J. C. Cummiskey
   Box 555171
   Camp Pendleton, CA 92055-5080

8. Chairman, Code EC/Hu 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

9. Professor Gary Hutchins, Code EC/Hu 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

10. Professor Harold Titus, Code EC/Ts 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

11. CAPT J.R. Powell, USN, Code CC/Pw.. 1
Undersea Warfare Academic Group
Naval Postgraduate School
Monterey, CA 93943-5121

12. Commander, Electronic Combat Wing, U.S. Pacific Fleet...................................... 1
N8 Requirements
Attn: Capt Olmstead
3730 N. Charles Porter Ave.
Oak Harbor, WA 98278-7500

13. Steven P. Jones 4
204 Nancy Ave.
N. Linthicum, MD 21090